650 DATA PROCESSING SYSTEM BULLETIN

IMMEDIATE ACCESS STORAGE, INDEXING REGISTERS, AUTOMATIC FLOATING-DECIMAL ARITHMETIC, AND MAGNETIC TAPE

This bulletin on additional features obsoletes:

1. Form 224-6258, 650 MDDPM Additional Features
2. Form 224-6265, 650 Manual of Additional Features
3. Form 328-7983, 650 Bulletin 5, pp. 8 and 9
4. Form 328-7675, 650 Bulletin 7
5. Form 328-7884, 650 Bulletin 10, pp. 69 and 70
6. Form 328-0314, 650 Bulletin 14
7. Form 328-0410, 650 Bulletin 15, pp. 111-115, 123-124, 127-128.
8. Form 328-0418, 650 Bulletin 19

The building-block principle employed in the IBM 650 Data Processing System enables it to meet the individual requirements of the intermediate area of data processing.

A 650 system can be tailored to individual needs because it can consist of varying numbers and types of units (Figure 1). Starting with a basic card operating system, consisting of the IBM 650 Console, the IBM 655 Power Unit and the IBM 533 Card Read Punch, the system can build up to a total of 31 units. The chart

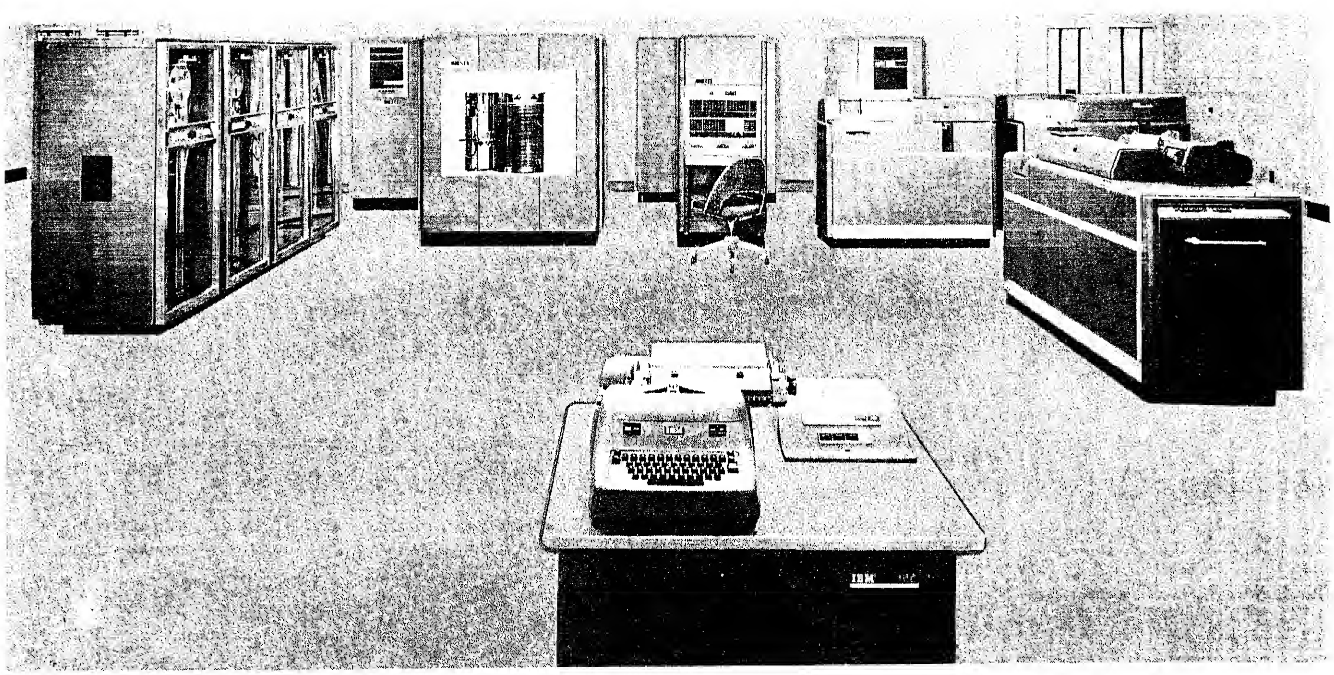


Figure 1. IBM 650 Data Processing System with Magnetic Tapes and RAMAC

(Figure 2) is a summary of units that can be used in a 650 system.

The IBM 652 Control Unit provides the necessary power and control circuitry for tape units, disk storage units, and inquiry stations.

The IBM 653 Storage Unit contains any or all of the following:

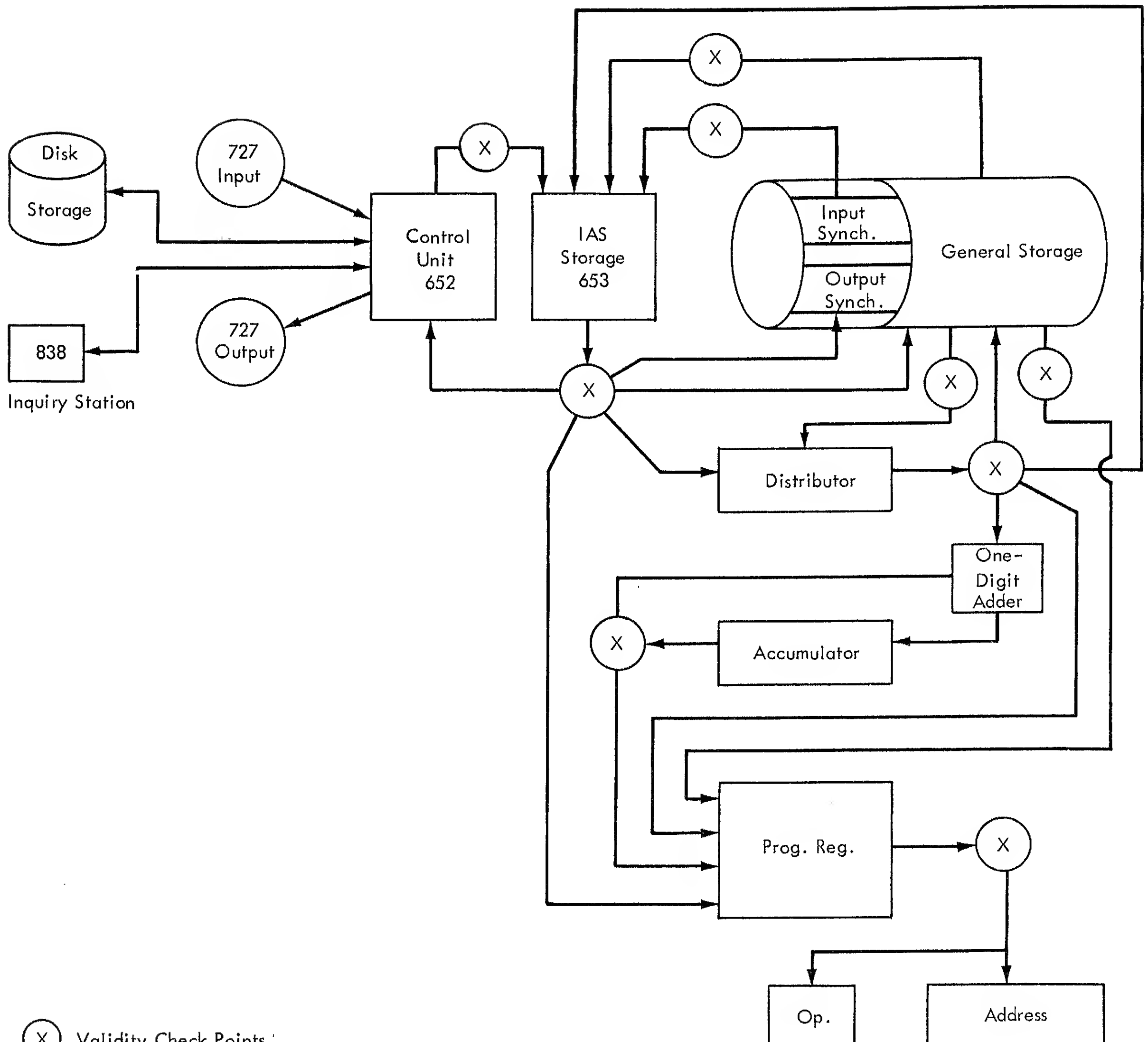
1. *Immediate Access Storage.* Increases the operational scope of the 650 and is a link connecting the magnetic-tape units or disk-storage units to the 650 system.
2. *Indexing Registers.* Can automatically modify addresses, and facilitate the necessary address arithmetic. They can also be used for accumulating small totals (up to 9999), retaining group multipliers, or as immediate-access storage devices.
3. *Automatic Floating-Decimal Arithmetic Device.* A means of performing automatic decimal-point alignment on numbers having a large range of values and written in floating-decimal form.

The IBM 727 Magnetic Tape Unit provides high-speed input and output for the IBM 650 Data Processing System. It also is a compact storage medium for large files of data. A single reel of tape, 10½ inches in diameter, can store the same amount of data as 56,000 fully punched IBM cards.

The flow paths in Figure 3 are the flow of information throughout the 650, as well as to and from Immediate Access Storage, Magnetic Tape Units, Disk Storage and the Inquiry Stations.

Unit	Name		Max. No. of Units	No. of Assoc. Operation Codes
Processing Units	IBM 650 Console (with 20,000 digits of storage) IBM 655 Power Unit		1 1	42
Card Input and Card Punch or Print Output	IBM 533 Card Read Punch Unit IBM 537 Card Read Punch Unit IBM 407 Accounting Machine IBM 543 Card Reader IBM 544 Card Punch	Input & Output Input & Output Input & Output Input Only Output Only	Up to three inputs and outputs	9
Paper Tape Input	IBM 46 Tape to Card Punch IBM 47 Tape to Card Printing Punch	Input Only Input Only		
Character Sensing Input	IBM 1210 Model 2 Sorter Reader	Input Only	1	1
Additional Features	IBM 653 Storage Unit (needed for Magnetic Tapes or Disk Storage) Contains any or all of the following: 1. 600 digits of immediate access core storage 2. 3 four position indexing registers 3. Automatic Floating Decimal Arithmetic		1	 5 18 7
Control	IBM 652 Control Unit (needed for magnetic tapes or Disk Storage)		1	--
Magnetic Tapes	IBM 727 Magnetic Tape Unit		6	10
IBM RAMAC®	IBM 355 Disk Storage Unit Model 1 (each unit stores 6 million digits) or, IBM 355 Disk Storage Unit Model 2 (each unit stores 12 million digits)		4	3
Inquiry	IBM 838 Inquiry Station (IBM 652 Control Unit Required)		10	2
	TOTAL		31	97

Figure 2. Summary of Units used in the IBM 650 Data Processing System



X Validity Check Points

Figure 3. Schematic of Flow Paths

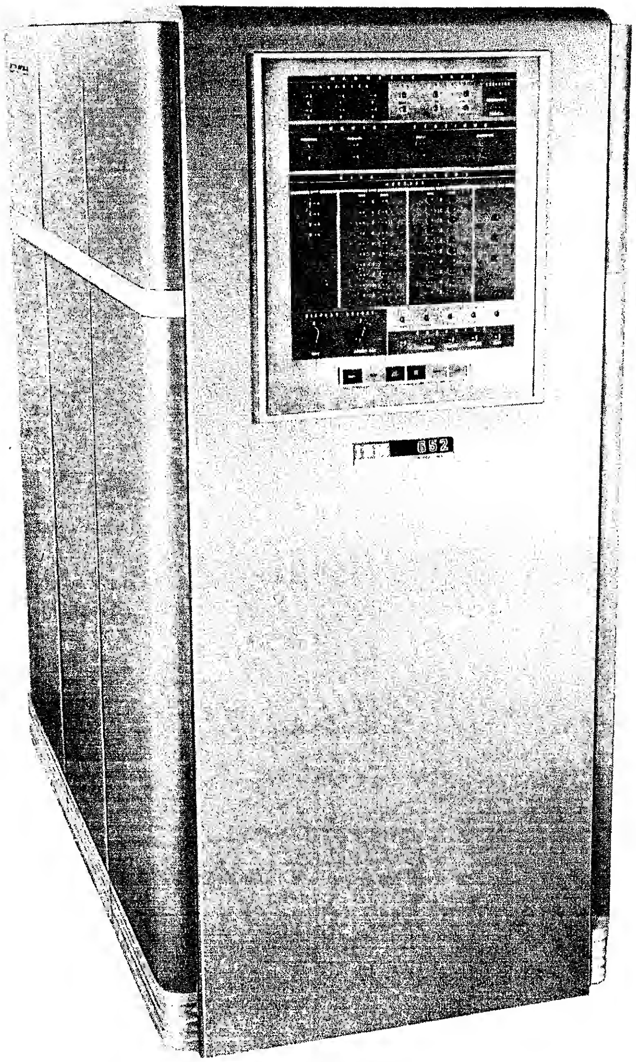


Figure 4. IBM 652 Control Unit

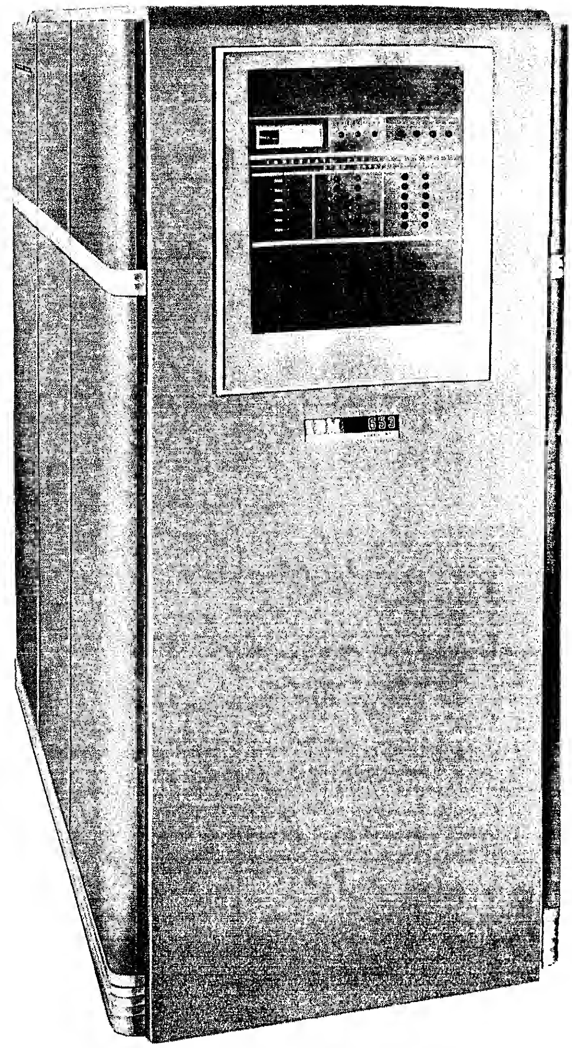


Figure 5. IBM 653 Storage Unit

IBM 652 Control Unit

The control unit provides the power and control for the IBM 727 Magnetic Tape Units (from one to six tape units). It also provides the control for as many as four IBM 355 Disk Storage Units, and as many as ten IBM 838 Inquiry Stations. Only one IBM 652 Control Unit can be used in an IBM 650 Data Processing System, and one IBM 653 Storage Unit must be used with this unit.

Various models of the 652 are available to meet specific needs for a data processing system.

IBM 653 Storage Unit

This unit is an integral part of IBM 650 Data Processing Systems using magnetic tapes and/or disk storage. Only one 653 can be used with a 650. When this unit is used with a magnetic tape or disk storage system, an IBM 652 Control Unit is also required.

Various models of this unit, to meet specific system needs, are available. This unit can contain:

1. 600 digits of immediate access storage
2. three 4-position indexing registers
3. automatic floating-decimal arithmetic.

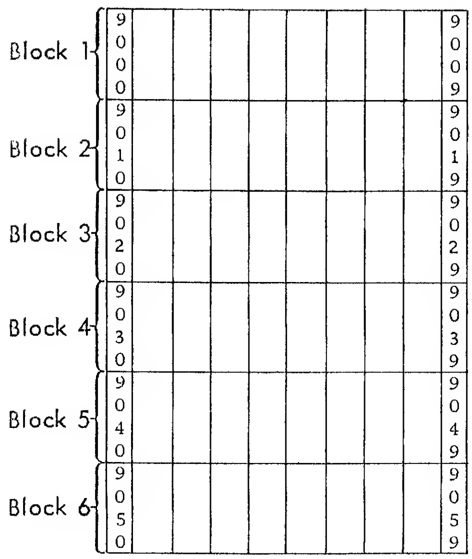


Figure 6. Immediate Access Storage

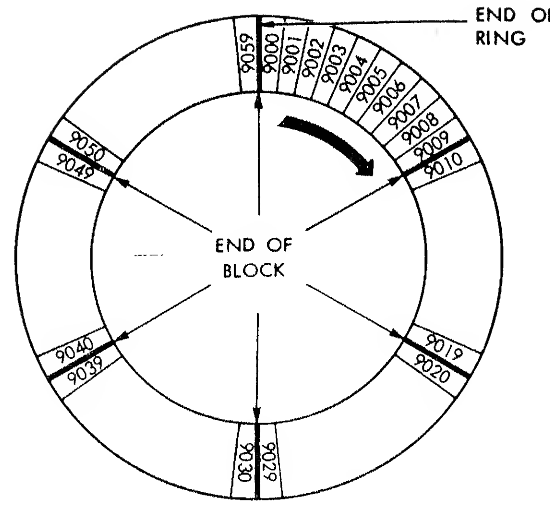


Figure 7. Immediate Access Storage Timing Ring

IMMEDIATE ACCESS STORAGE

Immediate Access Storage (IAS) is a 600-digit (60 words) immediate-access magnetic-core storage unit, and is located in the IBM 653 Storage Unit. Each of the words, 9000 through 9059, is addressable. Figure 6 shows a layout of IAS. Note that it is divided into six blocks of ten words each. Magnetic-core storage is a non-destructive read-out storage. This means that when information is read into IAS it remains for repeated use, until new information is read in to replace it. All data entering and leaving IAS is *validity-checked.*

IAS provides entry and exit of 60 words (600 digits) of information for reading and writing magnetic tapes and disk storage. It has a timing ring that keeps track of the word in IAS that is being operated upon.

Transfer of Data to and from IAS

Data can be transferred to IAS by:

1. Single-word transfers from the distributor or accumulator (store operations).
2. Multiple-word transfers from general storage.
3. Direct transfer from any input device, through an input synchronizer storage area.

Data can be transferred from IAS by:

1. Single-word transfers to the distributor, accumulator, or program register.
2. Multiple-word transfers to general storage.
3. Direct transfer to any output device, through an output synchronizer storage area.

Timing Ring

Immediate Access Storage uses a timing ring to keep track of core locations (Figure 7). A single operation code is associated with it.

27 SET *(Set IAS Timing Ring).* This instruction sets the timing ring at the IAS location specified by the D-address. Therefore, only IAS addresses (9000-9059) are valid D-addresses for this instruction. The 90xx addresses are *never* used in the I-address portion of this instruction.

The timing ring is set to enable a multiple-word transfer to be started with any word in IAS. The ring setting designates the first core-word affected by a multiple-word transfer. This instruction usually precedes a multiple-word transfer instruction (04, 05, 06, 07, 08, 09, 28, 29).

The timing ring automatically moves on multiple-word transfers (except transfer from input synchronizers). At the completion of a multiple-word transfer, the timing ring is at the location of the last word transferred, plus one. For example: last word transferred 9019 — timing ring is at 9020; last word transferred 9035 — timing ring is at 9036; last word 9059 — timing ring is at 9000, the beginning of IAS.

Single-Word Transfers

A single-word transfer to and from IAS is accomplished in the same manner as single-word transfer on the drum. The regular 650 operation codes can be used: load distributor, store distributor, reset-add lower, store lower, etc. In one-word transfers it is not

necessary to set the timing ring. For single-word transfers or for read instructions, the timing ring remains at the last location addressed in the instruction. Given the instruction:

65 9006 0303

the timing ring is set at 9006 at the end of the instruction. If, however, the instruction is:

65 9006 9002

the timing ring is set at 9002 at the end of the instruction.

Multiple-Word Transfer Instructions

From 1 to 50 words can be transferred between the drum and IAS by the multiple-word transfer instructions. The number of words transferred is dependent on:

1. The setting of the timing ring
2. The operation code used
3. The end of a band on the drum.

09 LDI *(Load IAS).* This instruction causes a transfer of from 1 to 50 words from the drum into IAS. The D-address of the LDI instruction specifies the location of the first word of the drum to be transferred into IAS. The location of the first word in IAS is determined by the setting of the timing ring. A SET instruction normally precedes an LDI instruction.

Once the transfer of information begins, it continues until location 9059 has been loaded from the drum, or the contents of the last location of the selected drum band are loaded into a location in IAS. The timing ring is set to the next consecutive location of IAS. For example:

SET 27 9000 xxxx
LDI 09 1000 xxxx

The 50 words (1000 through 1049) are loaded in locations 9000 through 9049 in IAS. The transfer is ended by an end-of-band condition (1049). The timing ring is left at 9050. Assume these instructions are followed by:

LDI 09 1050 xxxx

Upon execution of this instruction, words 1050 through 1059 would load into 9050 through 9059. The transfer is ended by an end-of-IAS condition (9059). The timing ring is left at 9000.

29 STI *(Store IAS).* The STI instruction causes the transfer of from 1 to 50 words from IAS locations to the drum. The D-address of the STI instruction specifies the first drum location loaded from IAS. The first word loaded on the drum is taken from the location to which the timing ring is set. Once begun, the information transfer continues until either the last location of a drum band is loaded from IAS, or the word in location 9059 has been moved to the drum. Upon termination of the transfer, the timing ring is set to the next consecutive location of IAS. For example:

SET 27 9000 xxxx
STI 29 1102 xxxx

This results in words 9000 through 9047 being stored in locations 1102 through 1149. The transfer is ended by an end-of-band condition (1149). The timing ring is left at 9048. Assume these two instructions are followed by:

STI 29 1200 xxxx

Upon execution of this step, words 9048 through 9059 are stored in locations 1200 through 1211. The transfer is ended by an end-of-IAS condition (9059). The timing ring is left at 9000.

08 LIB *(Load IAS Block).* The LIB instruction causes a maximum of 10 words to be transferred from the drum to IAS locations. The first of the words from general storage is specified by the data address of the instruction. The first word of IAS is determined by the position of the timing ring when the 08 command is executed. The loading operation is completed when the end of a band in general storage is reached (0049, 0099, etc.), or when the end of one of the blocks in IAS is reached (9009, 9019, 9029, etc.).
For example:

SET 27 9015 xxxx
LIB 08 1847 xxxx

This results in words 1847 through 1849 being transferred to locations 9015 through 9017. The transfer is ended by an end-of-band condition (1849). The timing ring is set to the next consecutive location of IAS or 9018. Assume these instructions are followed by:

LIB 08 1240 xxxx

Upon execution of this step words 1240 and 1241 are transferred to 9018 and 9019. The transfer is ended by an end-of-block condition (9019). The timing ring is at 9020.

28 SIB *(Store IAS Block).* The SIB instruction causes a maximum of 10 words from IAS locations to be moved into general storage. The first location of general storage loaded from IAS is specified by the D-address of the SIB instruction. The first word transferred from IAS is determined by the setting of the timing ring.

Once started, the information transfer is ended either when the contents of the last location of the IAS block are moved into general storage, or when the last location of a drum band is loaded from IAS.

Upon termination of the information transfer, the timing ring is set to the next consecutive location of IAS. For example:

SET 27 9015 xxxx

SIB 28 1004 xxxx

Upon completion of the above instructions, words 9015 through 9019 are stored in locations 1004 through 1008. The transfer is ended by an end-of-block condition (9019). The timing ring is at 9020. If this is followed by:

SIB 28 1147 xxxx

then words 9020 through 9022 are stored in 1147 through 1149. The transfer is ended by an end-of-band condition. The timing ring is left at 9023.

Figure 8 illustrates schematically the transfer of fewer than 10 words to/from IAS and drum using an end of IAS block condition for terminating the transfer.

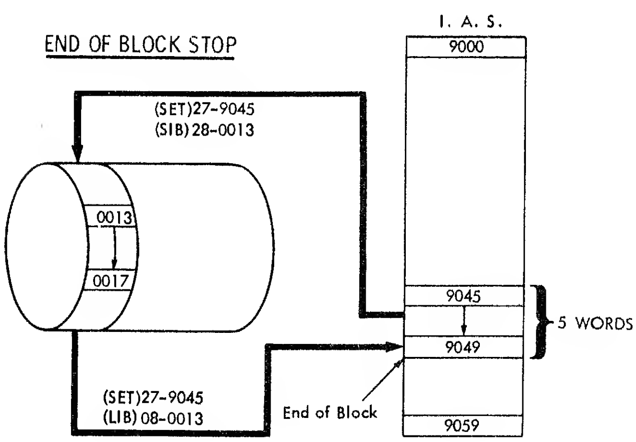


Figure 8. Transferring Less than Ten Words — End of Block Stop

Using IAS with Input/Output Units

Read Codes 70, 72, 73, 75, 76, 78. Ten or less words can be transferred directly to IAS from an input synchronizer. The data address of the read instruction determines the first word in IAS to be filled. The end of the 10-word block stops the transfer. For example: 70 9000 transfers 10 words from input synchronizer 1, to IAS locations 9000-9009. *The timing ring, at the end of the read instruction, is at 9000.* The instruction 70 9004 transfers words 1-6 from input synchronizer 1, to IAS locations 9004-9009 and locations 9000 to 9003 are left undisturbed. *The timing ring, at the end of this read instruction, is at 9004.* Any of the six 10-word blocks in IAS can be used as a read area.

As the data is transferred, it is checked for validity. This validity-check detects double punches, blank columns, and incorrectly wired word-size emitters. In a load card, all words reading into IAS must be punched and signed.

If the card is not a load card, the machine goes to the location specified by the I-address for the next instruction. If this I-location is an IAS location, the timing ring is moved from the D-address to the I-address location during the I-half cycle, and processing continues.

Write Codes 71, 74, 77. Ten or less words can be transferred directly from IAS to the output synchronizer.

A WR instruction sets the timing ring to the IAS location specified by the D-address of the instruction and transfers up to 10 words from IAS to the output synchronizer. The contents of the location specified by the D-address are read into word 1 of the output synchronizer. The timing ring is then set to the next location; and the contents of that location are read into word 2 of the output synchronizer. This process continues until the last location of the IAS block (9009, 19-29-39-49-59) has been transferred into the output area. The timing ring is then set to the first location of the following block, and all words in the output synchronizer, not loaded from IAS, are filled in with zeros. As the data is transferred from IAS to the output synchronizer, it is checked for validity.

Figure 9 illustrates schematically the transfer of five words to/from IAS and the input/output device.

The data address of the read instruction (70 9005) sets the timing ring and determines the first word in IAS to be filled. The end of the 10 word block (9009) stops the transfer.

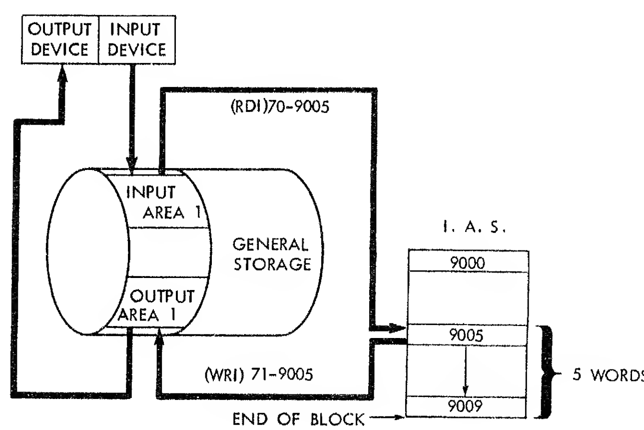


Figure 9. Transferring to/from IAS and Input/Output Device

The data address of the *write* instruction (719005) sets the timing ring at 9005. The contents of 9005 are read into word 1 of the output area, contents of 9006 into word 2, etc., until the contents of 9009 (end of IAS block) has been read into word 5. The end of IAS block stops the transfer.

IAI (Immediate Access Impulse). This impulse is available on the control panels of the input-output devices. It is used for format control when less than ten words are transferred from IAS to the output synchronizer area. For example, when less than ten words of data are transferred from IAS to an output-synchronizer, zeros are automatically inserted in the remaining word locations. These zeros can be eliminated from the output card by using selectors controlled from the early impulse emitted by the IAI hub.

Using IAS for Table Look-Up

When a TLU 84 instruction is executed and the D-address is an IAS location, the timing ring is set to the location specified by the D-address, and a table search starts at that location. Although the TLU can begin at any IAS location, the search doesn't start until the first drum words (0000, 0050, 0100, etc.), pass the read heads.

Beginning with the contents of the location specified by the D-address, the contents of each location are successively compared with the contents of the distributor until the value in the IAS location is either equal to or greater than the value in the distributor. When this condition occurs, the address of that location is inserted into the D-address portion (digit positions 5-8) of the word in the lower accumulator.

(NOTE: During the table search, all words in IAS and the word in the distributor are considered to be positive).

If no word is found to be greater than or equal to the word in the distributor, a storage selection-error stop occurs when the machine attempts to search location 9060 (Invalid Address). To prevent such stops from occurring, the last word of an IAS table should be 99 9999 9999 +; or prior to the beginning of TLU, the last word should be compared with the TLU argument to ensure that the argument is smaller than the last word in the table. Upon completion of the search, the timing ring is left at the last location searched, plus one.

Optimum Programming Using 90xx Addresses

When 90xx locations are used as D and I-addresses of program instructions, they are always the equivalent of an optimum drum location provided the Immediate Access Storage Interlock (IAS-I) is not set (see *Timing Considerations*). Therefore, as long as successive addresses are for core storage locations, optimizing is automatic.

In practice not all addresses are for core storage locations, but shift back and forth between IAS and the drum. When addresses are to shift back to the drum after reference to IAS, the programmer must ensure that an optimum drum location is chosen. To do this, the programmer need only keep track of the *equivalent* drum address when core locations are being used (See *Optimum Programming Chart,* Form X24-6219).

The following table shows the equivalent 90xx address when using a drum address for an instruction or data address.

Equivalent 90xx Address

1. When n (instruction address) is a drum address,

LOC OF INST n	OPERATION	DATA d	INSTRUCTION
EVEN	All add or subtract	n + 3	n + 7
ODD	in accumulator	n + 3	n + 8
EVEN OR ODD	Load Distributor (69)	n + 3	n + 6

2. When d (data address) is a drum address

All additions or subtractions	Even	d + 5 #
in accumulator	Odd	d + 4 #
Load Distributor (69)	Even or Odd	d + 3

Add 2 if a complement cycle is taken

In the following example it is necessary to optimize the drum D-address of the third instruction.

LOCATION OF INSTRUCTION	INSTRUCTION OP	DATA	INSTRUCTION	OPERATION ABBRV.
0001	60	9010	9049	RAU
9049	10	9011	9050	AUP
9050	11	XXXX		SUP

In the first instruction, 9010 is equivalent to drum location 0004 (n + 3) and 9049 is equivalent to 0009 (d + 5 because the equivalent D-address is even). In the second instruction 9011 is equivalent to 0012 (n + 3) and 9050 is equivalent to 0017 (d + 5). Therefore, with 9050 equivalent to 0017, the optimum drum location for the D-address of the third instruction is 0020 (n + 3).

In the following example it is necessary to optimize the drum I-address of the second instruction.

LOCATION OF INSTRUCTION	INSTRUCTION			OPERATION ABBRV.
	OP	DATA	INSTRUCTION	
0001	04	8010	0006	RTN
0006	60	9051	XXXX	RAU

In this example it is not feasible to optimize the I-address of the second instruction because *no* equivalent drum address can be assigned to 9051. At first glance it might appear that 9051 is equivalent to 0009 (n + 3). However, the first instruction sets the IAS-I until tape reading is completed. How long the IAS-I remains set depends on the length of the tape record. Therefore, programs should be planned not to refer to IAS until the IAS-I is removed. See *IAS Sequence Chart,* Figure 10.

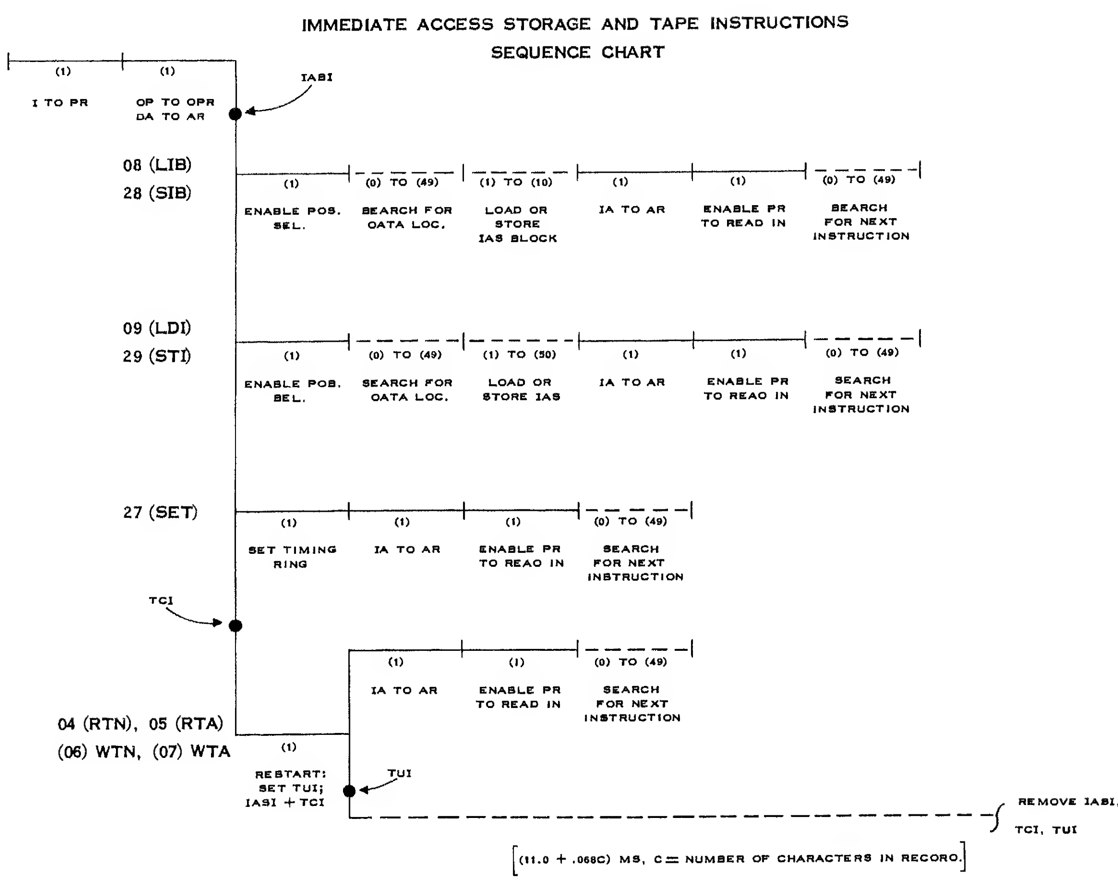


Figure 10. Immediate Access Storage Sequence Chart

Indexing Registers

In 650 programming, many problems require that the same operations be performed repetitively, with a change only in the data or instruction address. Modifying these addresses each time a repetitive operation is performed requires several program steps, plus additional drum locations which must be set aside for this use.

A group of three Indexing Registers (IR) located in the IBM 653 Storage Unit can be installed to modify addresses automatically. This means that less instructions are needed, which in turn means that additional drum storage space is made available. This provides for faster execution of a program and an overall simplification of programming effort.

Although the primary use of indexing registers is to modify addresses, they can be used as small accumulators, or as small immediate-access storage devices. Each indexing register has four digit positions and the associated algebraic sign. Operation codes associated with index registers permit factors to be inserted by reset-add or reset-subtract operations; it is possible to test each indexing register for a zero or non-zero, or for a plus or minus condition. Each indexing register is addressable so that its contents can be used as a factor in other operations. On the 650 Console, only the computer-reset key resets the indexing registers to zero.

Figure 11 is a schematic illustration of data flow—using indexing registers.

Indexing Register Addresses

Addresses assigned to the indexing registers are:

Indexing Register	*Address*
IRA	8005
IRB	8006
IRC	8007

Address Modification

The primary use of indexing registers is to modify addresses automatically by adding (algebraically) the contents of an indexing register to an address. The indexing register can contain either positive or negative values, making it possible to modify addresses by tagging the instruction. Both data addresses and instruction addresses can be modified by the contents of any indexing register or each by a different indexing register. Operation codes cannot be modified automatically by indexing registers.

Tagging

It is necessary to tag each address by an indicator so that the 650 may know which indexing register is to be added to the address. Addresses 2000 through 7999 have been reserved for this purpose. A basic drum address is one in the range 0000-1999. In order to tag drum addresses, either 2000, 4000, or 6000 is added to indicate that the contents of indexing register A, B, or C, respectively, are to be added to the basic drum address. Tagging immediate-access storage addresses is accomplished by adding 200, 400, or 600 to the IAS address, to indicate the use of indexing register A, B, or C, respectively.

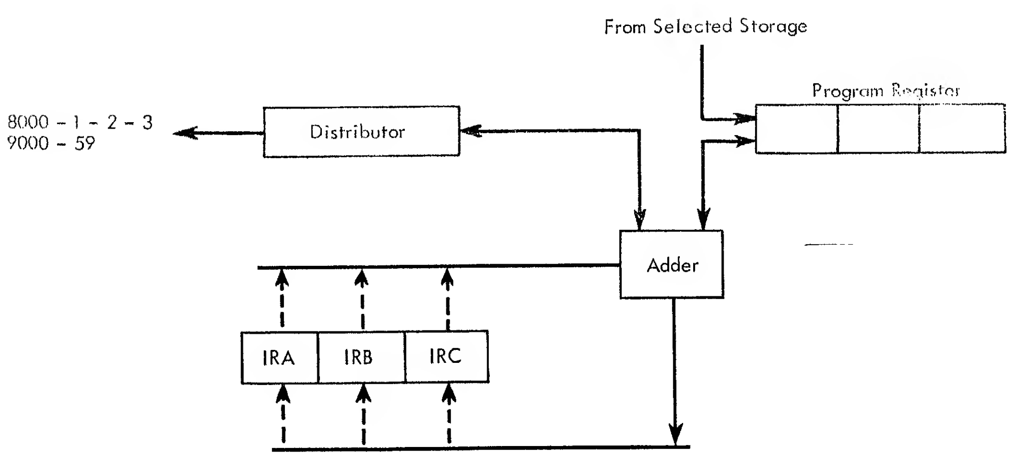


Figure 11. Indexing Registers Data Flow Schematic

It is not possible to tag the following addresses with indexing registers:

8000	Console
8001	Distributor
8002; 8003	Accumulators
8005-8007	Indexing Registers
8010-8015	Magnetic Tape Units

However, the developed address after modification by an indexing register can be any one of the preceding, if it is meaningful to the operation.

Instruction Indexing

When an instruction is read into the program register of an IBM 650 equipped with indexing resigters, these steps are executed:

1. The D- and I-addresses are checked for tagging.
2. If the D-address is tagged, it is modified by the appropriate indexing register.
3. If the I-address is tagged, the address is modified.

After a single modification of either or both addresses, the instruction is executed. At this point the data address must be meaningful to the operation called for, and the instruction address must be a valid storage address. If the indexing operation develops another tagged address, that address is considered invalid when read into the address register, and a storage-selection error-stop occurs.

In the following example assume that indexing register A contains 0300 + and the instruction 65 2760 xxxx is given.

1. When the instruction is read into the program register and tested, it is found that the data address is tagged.

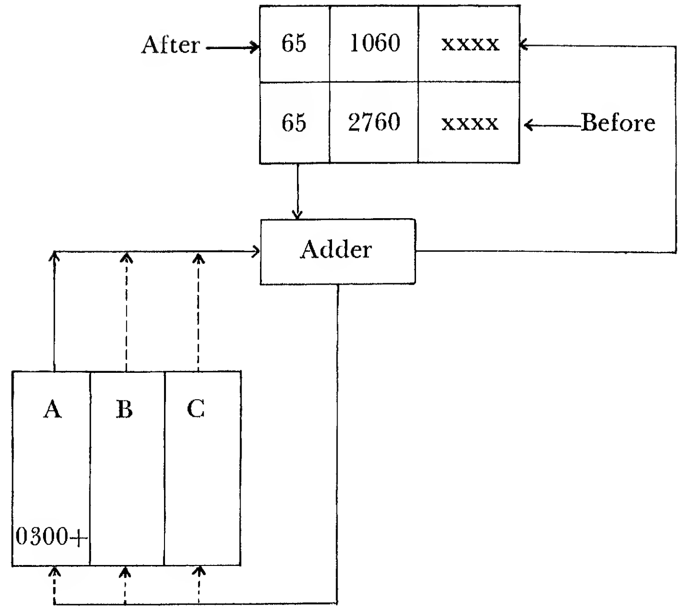
Program Register

65 2760 xxxx

Tag

2. The tag (2000) is a signal to the 650 to take a modification cycle and read out the contents of indexing register A (IRA).
3. A modification cycle is taken to remove the tag (2000) and add the contents of IRA to the D portion of the program register. This modification takes one word time.

Program Register



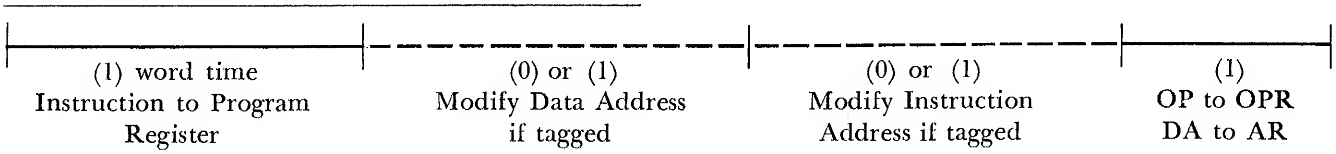
2760	
−2000	Minus tag
0760	
+0300	Contents of IRA
1060	Modified data address

4. The instruction is then executed to reset-add into the lower accumulator the contents of drum location 1060.

For another example assume that indexing register A contains 4000 + and the instruction xx 3999 xxxx is given. After modification by indexing register A the instruction reads xx 5999 xxxx, and xx 5999 is read into the operation register and address register. At this point a storage-selection-error stop occurs, because 5999 is an invalid address.

Optimum Timing

The following sequence of operations is applicable only to 650's equipped with indexing registers.



Developed Addresses

The developed address results after a drum or IAS address has been modified by the contents of an indexing register. The following table lists all meaningful addresses and the resulting addresses developed if a drum or IAS address is tagged.

Meaningful Address	Developed Address
0000-1999	0000-1999
2000-3999	0000-1999 + contents of IRA
4000-5999	0000-1999 + contents of IRB
6000-7999	0000-1999 + contents of IRC
8000-8003	8000-8003 Arithmetic Units
8005-8007	8005-8007 Index-Address
8010-8015	8010-8015 Tape Units
9000-9059	9000-9059 IAS
9200-9259	9000-9059 + contents of IRA
9400-9459	9000-9059 + contents of IRB
9600-9659	9000-9059 + contents of IRC

If an invalid address is developed through address modification, the machine stops and the storage-selection light turns on.

Indexing Operation

Address modification is accomplished by adding the contents of an indexing register to a basic address. See example 2, Figure 12. If the contents of the indexing register is positive and the resulting effective address exceeds 9999, the carry is lost and only the four low-order digits of the sum are kept as the effective address. In example 9, Figure 12, the data address 9210 calls for the adding indexing register A to 9010. This results in a total of 10993. However, only the four low-order digits are kept as part of the instruction. The digit 1 (carry) is lost. The machine does not stop because of this overflow.

If the contents of the indexing register are negative, the address is modified by subtraction. In example 7, Figure 12, the I-address (9627) is modified by the contents of indexing register C. This means that a minus 0015 is added to 9027, with a result of 9012, which is the indexed I-address. Because subtraction is accomplished by adding the 10's complement, a carry always occurs when the difference is positive. In the example just described, the contents of index register C (minus 0015) is subtracted from 9027 by adding the 10's complement. The sum of 9027 plus 9985 (10's complement of 0015) is 19012. Because the total exceeds 9999, the carry is lost and only the four low-order digits (9012) are kept as the effective address.

	Actual Instruction	*Contents of I R*			*Indexed Instruction*	*Remarks*
		A	*B*	*C*		
1.	65 0123 0124	0223+	0075 −	0062+	65 0123 0124	No Indexing
2.	65 2123 0124	0223+			65 0346 0124	Index D by A
3.	65 0123 6124			0062+	65 0123 0186	Index I by C
4.	65 4123 4124		0075 −		65 0048 0049	Index D and I by B
5.	65 4123 6124		0075 −	0062+	65 0048 0186	Index D by B and I by C
6.	65 9215 8002	0013+			65 9028 8002	Index D by A
7.	65 0123 9627			0015 −	65 0123 9012	Index I by C
8.	65 4015 0124		2345+		65 2360 0124	2360 causes storage selection error.
9.	65 9210 0124	1983+			65 0993 0124	D exceeds 10,000; carry is lost.
10.	65 0123 4124		7878+		65 0123 8002	I becomes 8002.
11.	65 9615 9218	1011 −		0015 −	65 9000 8007	I becomes 8007.
12.	65 2123 0124	1011 −			65 9112 0124	D becomes "negative." Complement 9112 causes storage selection error.
13.	65 2123 0124	1111 −			65 9012 0124	D becomes "negative." Complement is meaningful, however.
14.	04 2000 xxxx	8010			04 8010 xxxx	Valid Tape address.

Figure 12. Examples of Address Modification

If indexing by subtraction results in a negative address, the complement result is not reconverted. This may result in a storage selection error if the complement is not a meaningful address. In example 12, Figure 12, the indexed address (9112) is negative. Because it is an invalid address a storage-selection error occurs. In example 13, Figure 12, the indexed address is negative, but because the complement is a valid address, no error is indicated.

The other examples in Figure 12 illustrate some of the various ways indexing registers can be used.

IR Arithmetic Operations

Four operation codes are associated with each IR, to add, subtract, reset-add, or reset-subtract data into each register.

Operation Codes

	IRA	*IRB*	*IRC*
Add	50 (AXA)	52 (AXB)	58 (AXC)
Subtract	51 (SXA)	53 (SXB)	59 (SXC)
Reset Add	80 (RAA)	82 (RAB)	88 (RAC)
Reset Subtract	81 (RSA)	83 (RSB)	89 (RSC)

When index registers are used as accumulators, they are similar to the upper and lower accumulator except that:

1. they are smaller (4 positions)
2. they do not indicate when an overflow occurs
3. they can accept data only from positions 1-4 of another immediate access storage device, or positions 5-8 of the program register
4. they cannot accept data directly from a drum location
5. they subtract by the 10's complement method, and a complement result is reconverted.

If the contents of an indexing register is to be operated on by an IR arithmetic code, the IR tag address 2000, 4000, 6000 must be used. Machine addresses 8005-6-7 must never be used. To add the contents of IRB into IRA the instruction would be: 50 4000 xxxx.

Using Indexing Registers as Accumulators

Data from 8000-1-2-3, 9000-59. The four low-order positions of data stored in any immediate access storage device: upper or lower accumulator (8002, 8003), the distributor (8001), the console switches (8000), or IAS (9000 to 9059), can be added into any index register with an IR arithmetic operation code.

Instruction	Data		Contents of IRA	
	Loc	Contents	Before	After
AXA 50 8001 xxxx	8001	139423 8621+	0000+	8621+
RAA 80 9021 xxxx	9021	006123 1426−	2713+	1426−
SXA 51 8000 xxxx	8000	006123 1426+	2426+	1000+
SXA 51 8002 xxxx	8002	152619 8712+	2426+	6286−

Drum Addresses. Data stored in drum addresses 0000 to 1999 cannot be added or subtracted directly into index registers by the IR arithmetic codes. When any number between 0000 and 1999 appears as the data address of an IR arithmetic operation code, the number itself is placed in the specified IR. The sign of this number is always treated as plus, because the program register carries no sign. This is a convenient way of placing factors in the index register with one instruction, and doesn't require another storage location to store the factor itself. It also affords a simple way of adding or subtracting 1 to count a program loop.

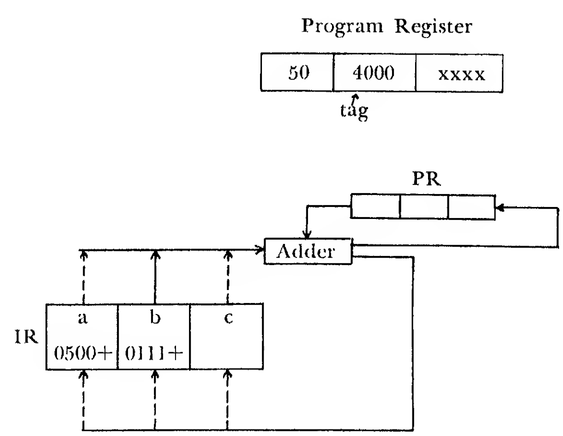
Instruction	Contents of IRA	
	Before	After
AXA 50 0126 xxxx	1111+	1237+
RAA 80 0126 xxxx	1111+	0126+
SXA 51 0126 xxxx	1226+	1100+
SXA 51 0126 xxxx	0000+	0126−

Arithmetic Operations Between Indexing Registers

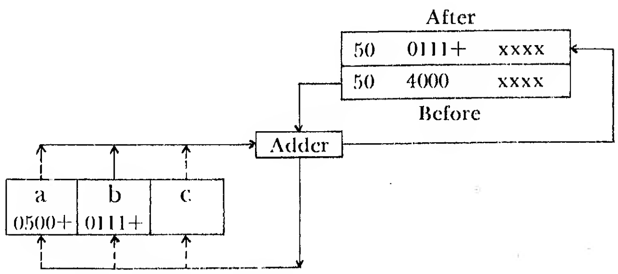
When data are to be added or subtracted from one indexing register to another, the data address must be the tag address of the IR containing the data to be added or subtracted. This is necessary because the data flow is similar to data flow when an instruction is indexed, but in this case the D portion of the program register acts as intermediate storage for one of the factors to be added.

In example 2, this sequence takes place when the instruction 50 4000 xxxx is placed into the program register:

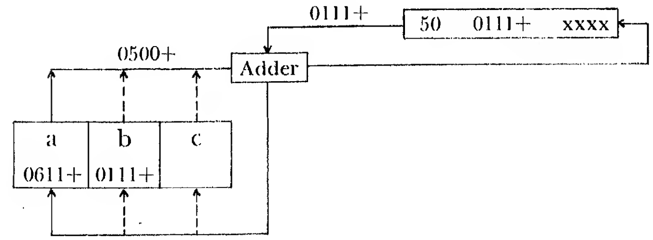
1. a test of the program register determines that the D-address is tagged;
2. the tag 4000 is a signal to the 650 to take a modification cycle and read out the contents of IRB;



3. a modification cycle is taken to remove the tag (4000) and add the contents of IRB to the D portion of the program register;



4. the instruction is executed to add into IRA what is stored in the D portion of the program register.



Program Register Before	IRA Before	IRA After	IRB Before	IRB After	Program Register After
1. AXA 50 2000 xxxx	0500+	1000+			50 0500 xxxx
2. AXA 50 4000 xxxx	0500+	0611+	0111+	0111+	50 0111 xxxx
3. AXB 52 2000 xxxx	9200–	9200–	0500+	1300+	52 0800 xxxx
4. AXB 52 2000 xxxx	2001+	2001+	0500+	2501+	52 2001 xxxx

Example 4 results in a machine stop with a storage selection light, because the developed address is invalid.

When any number from 0001 to 1999 is added to the tag address in the data-address portion of an indexing-register arithmetic-operation code, this number adds or subtracts into the IR affected, as well as the contents of the IR called for by the operation code.

In example 2, this sequence takes place when the instruction 50 4265 xxxx is placed into the program register:

1. a test of the program register determines that the D-address is tagged to read out IRB;
2. a cycle is then taken to remove the tag and algebraically add the contents of IRB to the D portion of the program register;
3. the result (0265 + 9889 = 0154 +) is stored back in the D portion of the program register;
4. the instruction 50 0154 xxxx is then executed to add into IRA the data stored in the D portion of the program register. IRA then contains (0500 + 0154) 0654.

Program Register Before	IRA Before	IRA After	IRB Before	IRB After	Program Register After
1. AXA 50 2156 xxxx	0500+	1156+			50 0656 xxxx
2. AXA 50 4265 xxxx	0500+	0654+	0111–	0111–	50 0154 xxxx
3. AXB 52 3456 xxxx	0000+	0000+	0000+	1456+	52 1456 xxxx
4. SXA 51 4250 xxxx	0500+	0375+	0125–	0125–	51 0125 xxxx
5. SXA 51 9207 xxxx	0050+	7940+	Loc 9007-1234567890		51 9007 xxxx
6. SXA 51 0001 xxxx	0500+	0499+			51 0001 xxxx
7. RAA 80 0000 xxxx	1234+	0000+			80 0000 xxxx
8. RAA 80 1520 xxxx	1234+	1520+			80 1520 xxxx
9. RAA 80 4000 xxxx	1234+	0175+	0175+	0175+	80 0175 xxxx
10. RAA 80 4525 xxxx	1234–	0700+	0175+	0175+	80 0700 xxxx
11. RSA 81 0002 xxxx	0069–	0002–	0150–	0150–	81 0002 xxxx
12. RSA 81 4002 xxxx	0069–	0152–	0150+	0150+	81 0152 xxxx

IR Branching Operations

Testing the indexing register to make logical decisions is done by the use of branch codes. Two branch operation codes are associated with each of the indexing registers to test them individually for a minus condition and for a non-zero condition.

OPERATION CODES

	IRA	IRB	IRC
Branch on non zero	40 NZA	42 NZB	48 NZC
Branch on minus	41 BMA	43 BMB	49 BMC

The data address (branch address) or the instruction address of an IR branch code can be any valid machine address (Drum, IAS, IR, arithmetic unit). If the next instruction is taken from an indexing register, it is treated as a no-op code, and the instruction following the no-op code is taken from the address specified in the index register.

The combination of IR arithmetic codes and IR branch codes permits simpler programming and reduced compute time, for counting program loops and branching into another part of the program when a pre-determined condition is reached. The upper and lower accumulator remain free to be used as desired for the arithmetic execution of the loop.

EXAMPLE: A factor in an input card determines how many times a particular program loop must be executed. The factor is read into positions 1 to 4 of location 0006, and position 5 to 10 can contain any other input data.

```
1502      LDD   69   0006      1503 }  put factor in IRA
1503      RAA   80   8001      1504 }
1504  [                     ]  1505 ]
                                       loop
1520  [                     ]  1521 ]
1521      SXA   51   0001      1522    Subtract 1
1522      NZA   40   1504      1523    Test loop count
1523    Next program step
```

40 NZA *Branch Non-Zero IRA.* If IRA contains zeros, the next instruction is taken from the location specified by the instruction address. If the contents of IRA are not zero, the next instruction is taken from the location specified by the data address.

41 BMA *Branch Minus IRA.* If the sign of IRA is plus, the next instruction is taken from the location specified by the instruction address. If it is minus, the next instruction is taken from the location specified by the data address.

42 NZB *Branch Non-Zero IRB.* If IRB contains zeros, the next instruction is taken from the location specified by the instruction address. If the contents of IRB are not zero, the next instruction is taken from the location specified by the data address.

43 BMB *Branch Minus IRB.* If the sign of IRB is plus, the next instruction is taken from the locations specified by the instruction address. If it is minus, the next instruction is taken from the location specified by the data address.

48 NZC *Branch Non-Zero IRC.* If IRC contains zeros, the next instruction is taken from the location specified by the instruction address. If the contents of IRC are not zero, the next instruction is taken from the location specified by the data address.

49 BMC *Branch Minus IRC.* If the sign of IRC is plus, the next instruction is taken from the location specified by the instruction address. If it is minus, the next instruction is taken from the location specified by the data address.

ARITHMETIC OPERATION CODES

In IR arithmetic operations positions 5-8 of an instruction (data address) contain the 4-digit number that is placed in the specified IR. The sign of this number is always treated as plus, because the program register carries no sign.

50 AXA *Add to IRA.* The absolute value (0000-1999) of the data address (or the effective data address modified by an IR) is added to the contents of IRA. The examples in Figure 13 show several ways in which 50 AXA can be used. The same general principles apply when using 52 AXB or 58 AXC except that indexing register B or C is used.

52 AXB *Add to IRB.* The data specified by the data address are added to the contents of IRC.

58 AXC *Add to IRC.* The data specified by the data address are addded to the contents of IRC.

	Actual Instruction	Indexed Instruction	Contents of I R A Before	Contents of I R A After	Contents of I R B	Contents of 8000-8003 or 9000-9059
1.	50 0001 0123	50 0001 0123	0500+	0501+		
2.	50 1623 0123	50 1623 0123	0500+	2123+		
3.	50 2000 0123	50 0500 0123	0500+	1000+		
4.	50 2156 0123	50 0656 0123	0500+	1156+		
5.	50 4000 0123	50 0111 0123	0500+	0611+	0111+	
6.	50 4265 0123	50 0154 0123	0500+	0654+	0111−	
7.	50 8002 0123	50 8002 0123	0500+	1611+		7777771111+
8.	50 9007 0123	50 9007 0123	0500+	7277−		1111117777−
9.	50 9407 0123	50 9004 0123	0500+	1734+	0003−	0202021234+
10.	50 2156 2123	50 0656 0623	0500+	1156+		

Figure 13. Operation Upon Indexing Registers

In Figure 13, examples 1 and 2, a constant (in the range) 0000-1999 is added to IRA.

In example 3, IRA is added to IRA.

In example 4, IRA is added to IRA and also added to another constant (in the range 0000-1999).

In example 5, IRB is added to IRA.

In example 6, IRB and a constant is added to IRA.

In example 7, the four low-order digits and sign of the lower accumulator are added to IRA.

In examples 8 and 9, the four low-order digits and sign are added to IRA.

Example 10 illustrates how addresses are modified before the operation is executed. Thus, the D (Data address) and I (Instruction address) are increased by 0500 before the contents of IRA are modified.

51 SXA *Subtract from IRA.* The data specified by the data address are subtracted from the contents of IRA. The examples in Figure 14 illustrate uses of this operation code.

53 SXB *Subtract from IRB.* The data specified by the data address are subtracted from the contents of IRB.

59 SXC *Subtract from IRC.* The data specified by the data address are subtracted from the contents of IRC.

In the following operations, the indexing register is reset to zero before the data are added or subtracted into it.

	Actual Instruction	Indexed Instruction	Contents of I R A Before	Contents of I R A After	Contents of I R B	Contents of 8000-8003 or 9000-9059
1.	51 0001 0123	51 0001 0123	0500+	0499+		
2.	51 2000 0123	51 0500 0123	0500+	0000+		
3.	51 4250 0123	51 0125 0123	0500+	0375+	0125−	
4.	51 8000 0123	51 8000 0123	0500+	7390−		1234567890+
5.	51 9207 0123	51 9057 0123	0050+	7940+		1234657890−

Figure 14. Indexing Registers Using 51 SXA Code

Actual Instruction	Indexed Instruction	Contents of I R A		Contents of I R C	Contents of 8000-8003 or 9000-9059
		Before	*After*		
1. 80 0000 0123	80 0000 0123	1234 −	0000+		
2. 80 1520 0123	80 1520 0123	1234 −	1520+		
3. 80 6000 0123	80 0175 0123	1234 −	0175+	0175+	
4. 80 6525 0123	80 0350 0123	1234 −	0350+	0175 −	
5. 80 9027 0123	80 9027 0123	1234 −	5021 −		001212345021 −

Figure 15. Indexing Registers Using 80 RAA Code

80 RAA *Reset Add to IRA*. IRA is reset to zero, and the data specified by the data address are added to it. The examples in Figure 15 illustrate uses of this operation code.

82 RAB *Reset Add to IRB*. IRB is reset to zero, and the data specified by the data address are added to it.

88 RAC *Reset Add to IRC*. IRC resets to zero, and the data specified by the data address are added to it.

81 RSA *Reset Subtract from IRA*. IRA is reset to zero, and the data specified by the data address are subtracted from it. The examples in Figure 16 illustrate uses of this operation code.

83 RSB *Reset Subtract from IRB*. IRB is reset to zero, and the data specified by the data address are subtracted from it.

89 RSC *Reset Subtract from IRC*. IRC is reset to zeros, and the data specified by the data address are subtracted from it.

SUMMARY OF INDEXING REGISTER ARITHMETIC CODES

The data specified by the data address of these operation codes fall into one of these categories:

1. If the data address used is in the range 0000-1999, or has been converted to this range by indexing, the absolute value, representing the data-address portion of the instruction, is added to the contents of one of the indexing registers.
2. If the data address refers to any of the following immediate-access storage devices 8000-1-2-3, 9000-59, or has been converted to one of these addresses by indexing, the four low-order positions of their contents are added to one of the indexing registers.

NOTE: The IR machine addresses 8005-6-7 cannot be used with any of these instructions, but IR tag addresses 2000, 4000, 6,000 can be used. However the result must be a valid address (0000-1999, 8001, 8002, 8003, 9000-9059).

Actual Instruction	Indexed Instruction	Contents of I R A		Contents of I R C	Contents of 8000-8003 or 9000-9059
		Before	*After*		
1. 81 1234 0123	81 1234 0123	0527+	1234 −		
2. 81 7015 0123	81 1927 0123	0527+	1927 −	0912+	
3. 81 7015 0123	81 0103 0123	0527+	0103 −	0912 −	
4. 81 9059 0123	81 9059 0123	0527+	2301+		0123012301 −

Figure 16. Indexing Registers Using 81 RSA Code

50AXA (Add to IRA). The data specified by the data address are added to the contents of IRA.

51SXA (Subtract from IRA). The data specified by the data address are subtracted from the contents of IRA.

52AXB (Add to IRB). The data specified by the data address are added to the contents of IRB.

53SXB (Subtract from IRB). The data specified by the data address are subtracted from the contents of IRB.

58AXC (Add to IRC). The data specified by the data address are added to the contents of IRC.

59SXC (Subtract from IRC). The data specified by the data address are subtracted from the contents of IRC.

80RAA (Reset Add to IRA). IRA is reset to zero, and the data specified by the data address are added to it.

81RSA (Reset Subtract from IRA). IRA is reset to zero, and the data specified by the data address are subtracted from it.

82RAB (Reset Add to IRB). IRB is reset to zero, and the data specified by the data address are added to it.

83RSB (Reset Subtract from IRB). IRB is reset to zero, and the data specified by the data address are subtracted from it.

88RAC (Reset Add to IRC). IRC is reset to zero, and the data specified by the data address are added to it.

89RSC (Reset Subtract from IRC). IRC is reset to zero, and the data specified by the data address are subtracted from it.

Use of IR Machine Addresses 8005-6-7

Used as instruction addresses, IR machine addresses 8005, 8006, 8007 can be used as instruction addresses of any instruction. When used this way, the next operation executed is a NO-OP (no operation), whose instruction address is the contents of the IR addressed.

For example:

Indexing Register A contains 1234+.

The instruction 65 0100 8005 is executed.

The lower accumulator is reset and the contents of 0100 are added in the lower accumulator. The instruction 00 0000 1234 is then executed, and it is a NO-OP—go to location 1234 for the next instruction.

Used as a data address, IR machine addresses 8005, 8006, 8007 can be used as data addresses of some instructions. Fifty-two of the ninety-six operation codes can use IR machine addresses. The following are the categories of machine operation codes that can use an IR machine address:

OPERATION CODES THAT CAN USE IR AS THE DATA ADDRESS

OPERATION DESCRIPTION	OPERATION CODES	REMARKS
No Operation Halt	00 01	Does not affect the function of the machine as these two instructions do not use the data address.
Upper and Lower Accumulator Arithmetic Operations	10, 11, 14, 15, 16, 17, 18, 19 60, 61, 64, 65, 66, 67, 68	The contents of IR are used as a factor in the operation. The four digits in the IR are the 4 low-order digits of a word that has 6 zeros in the high-order position. The distributor also contains this same data. Signs are manipulated just as any other word.
Floating Decimal Arithmetic	02, 32, 33, 34, 37, 38, 39	
Branch Operations	25, 26, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 54 90 to 99	The instruction executed is a NO-OP (no operation) whose instruction address is the contents of the IR addressed.
Shift Operations	30, 31, 35, 36	Only the units position is analyzed; therefore addresses of 8005, 6 or 7 cause shifting of 5, 6 or 7 places respectively.
Load Distributor	69	Loads with six zeros to the left and the contents of 8005, 6 or 7 in positions 1 to 4.

Operation Codes That Cannot Use IR As the Data Address

OPERATION DESCRIPTION	OPERATION CODE	DATA ADDRESS MUST BE:	If 8005-6-7 are erroneously used,
Tape read/write check Tape rewind Write tape mark Tape back space	03-04-05-06-07 55 56 57	727 Tape Unit Address 8010 to 8015	Stops on D half cycle Storage unit light Storage unit light Storage unit light
Load IAS or IAS block Store IAS or IAS block	08-09 28-29	Drum address	Storage unit light Storage selection and storage unit light
Set ring	27	IAS address	Stops on D half cycle
Store instructions	20-21-22-23-24	Drum or IAS address	Storage selection light
Index register add or subtract	50-51-52-53-58-59 80-81-82-83-88-89	IAS, LA, UA, Distrib. Console, and drum address*	Storage selection light
Input/output	70-71-72-73-74-75-76-77-78	Drum or IAS address	Stops on D half cycle
Table look-up	84	Drum or IAS address	Stops on D half cycle
Disk storage	85 86, 87	Any valid address 9000 -	 Stops on D half cycle
Inquiry; Reply	26, 79	Drum address	Stops on D half cycle

*When a drum address is used with these instructions, it does not refer to the contents of that drum location.

Meaningful Data Addresses

The data address for all 650 operation codes must be meaningful to the operation called for. Figure 17 shows the machine addresses possible for the data address of each operation code.

The following code is used:

Address	Code	Description
0000-1999	D	Drum
8000-8003	A	Arithmetic unit and console switches
8005-8007	I	Indexing Registers
8010-8015	T	Tapes
9000-9059	S	Immediate Access Storage

OPERATION CODE UNITS POSITION

OPERATION CODE TENS POSITION	0	1	2	3	4	5	6	7	8	9
0	NOP D A I T S	HLT D A I T S	UFA D A I S	RTC T	RTN T	RTA T	WTN T	WTA T	LIB D	LDI D
1	AUP D A I S	SUP D A I S			DIV D A I S	ALO D A I S	SLO D A I S	AML D A I S	SML D A I S	MPY D A I S
2	STL D S	STU D S	SDA D S	SIA D S	STD D S	NTS D A I S	BIN D	SET S	SIB D	STI D
3	SRT D A I T S	SRD D A I T S	FAD D A I S	FSB D A I S	FDV D A I S	SLT D A I T S	SCT D A I T S	FAM D A I S	FSM D A I S	FMP D A I S
4	NZA D A I S	BMA D A I S	NZB D A I S	BMB D A I S	NZU D A I S	NZE D A I S	BMI D A I S	BOV D A I S	NZC D A I S	BMC D A I S
5	AXA D A S	SXA D A S	AXB D A S	SXB D A S	NEF D A I S	RWD T	WTM T	BST T	AXC D A S	SXC D A S
6	RAU D A I S	RSU D A I S			DVR D A I S	RAL D A I S	RSL D A I S	RAM D A I S	RSM D A I S	LDD D A I S
7	RD1 D S	WR1 D S	RC1 D S	RD2 D S	WR2 D S	RC2 D S	RD3 D S	WR3 D S	RC3 D S	RPY D
8	RAA D A S	RSA D A S	RAB D A S	RSB D A S	TLU D S	SDS 9000	RDS 9000	WDS 9000	RAC D A S	RSC D A S
9	BDO D A I S	BD1 D A I S	BD2 D A I S	BD3 D A I S	BD4 D A I S	BD5 D A I S	BD6 D A I S	BD7 D A I S	BD8 D A I S	BD9 D A I S

Figure 17. Valid Data Addresses

Automatic Floating-Decimal Arithmetic

Many problems involve lengthy, complex calculations that require extensive analysis to determine the size and range of intermediate and final quantities. This analysis and the subsequent scaling of these quantities, frequently require a larger percentage of the total time to solve the problem than the actual calculations.

Floating decimal arithmetic eliminates this difficulty, because the numbers operated on are automatically scaled. The Automatic Floating-Decimal Arithmetic Device is located in the IBM 653 storage unit.

As an optional feature of the IBM 650, seven floating-decimal arithmetic operation codes are available to perform the operations of floating-add, floating-non-normalize add, floating-subtract, floating-add absolute, floating-subtract absolute, floating-multiply, and floating-divide. In this section of the bulletin the principles of floating-decimal arithmetic are explained briefly and the functions of the seven 650 floating-decimal operations are described.

In floating-decimal arithmetic in the 650, each word operated on is treated as an 8-digit number, and a 2-digit modified characteristic. When any floating decimal operation is executed, the numbers involved are interpreted:

$$(m, c) \approx \underbrace{xxxxxxxx}_{m}\underbrace{xx}_{c}$$

where *m* is the *mantissa*, and *c* is the *modified characteristic*.

The mantissa, m, consists of the left eight decimal digits. The location of the decimal point of the mantissa is fixed to lie immediately to the left of the extreme left digit. The sign of the number is always associated with the mantissa. A mantissa is called *normal* (or *normalized*) when its high-order digit is nonzero. If the high-order digit is zero, the mantissa is referred to as *non-normal* or *non-normalized.* (The exception to this rule is the floating decimal zero which is .0000000000.) Thus the range of the absolute value of the normal mantissa is

$$.10000000 \leq |m| \leq .99999999$$

The machine exponent may be explained: Since the decimal point always lies to the left of the eight digits, each mantissa must have associated with it some power of ten to specify the location of the decimal point of the original number represented by the floating decimal number. For example:

1. $123.45678 = (\ .12345678) \times 10^3$
2. $.00765438 = (\ .76543800) \times 10^{-2}$
3. $-.12348693 = (-.12348693) \times 10^0$
4. $-.00000070 = (-.70000000) \times 10^{-6}$

Because the sign of the floating decimal number is always associated with the mantissa, it cannot serve the function of indicating the sign of the power of ten. Obviously this creates a need for expressing the power of ten in some manner that will cause no conflict with positive or negative mantissas. This need is satisfied by algebraically adding fifty to the power of ten and using the resulting 2-digit number as the modified characteristic. Thus, the power of ten is transformed into the modified characteristic, c, which can assume a range of values from 00 to 99, inclusive, or:

$$00 \leq c \leq 99$$

Since c is the power of ten plus fifty, the power of ten has a range:

$$-50 \leq \text{power of } 10 \leq 49$$

Thus, normal floating decimal numbers can have a range of

$$\pm(.10000000 \times 10^{-50}) \text{ to } \pm(.99999999 \times 10^{49}), \text{ inclusive.}$$

The four examples shown previously are written as floating decimal numbers:

1.	$123.45678 = (\ .12345678) \times 10^3$	1234567853
2.	$.00765438 = (\ .76543800) \times 10^{-2}$	7654380048
3.	$-.12348693 = (-.12348693) \times 10^0$	1234869350–
4.	$-.00000070 = (-.70000000) \times 10^{-6}$	7000000044–

To summarize, a fixed decimal number, N, is represented by the floating decimal number

$$N = (\pm m) \times 10^{C-50} \approx (m,c)\pm$$

where m is the 8-digit mantissa, and c is the modified characteristic, and the sign is the sign of the mantissa, which is governed by the sign of N.

Operation Codes

Any floating-decimal operation that results in a zero mantissa forces a zero exponent. In all the operations, the lower half of the accumulator should be zero at the beginning of the operation. The lower half of the accumulator will be set to zero upon completion of each floating-decimal operation. A modified characteristic overflow ($c>99$) or underflow ($c<00$) from any floating-decimal operation causes the overflow

light on the console to be turned on, and the overflow trigger to be set. The overflow trigger can be interrogated if the overflow switch on the console is set to SENSE. If the overflow switch is set to STOP, all modified characteristic overflows and underflows cause the machine to stop. Dividing by zero causes an unconditional overflow stop; the setting of the overflow switch has no significance in this case. During all floating-decimal operations, the upper half of the accumulator is treated as an 8-digit position accumulator.

As a functional part of the execution of the operations, a left shift or carry from the high-order position of the lower, shifts or carries into the third position of the upper. Similarly, a right shift causes the digit in the third position of the upper to shift into the high-order position of the lower. After the arithmetic portion of each operation has been completed, a "5" is added or subtracted in the high-order position of the lower half of the accumulator to round the result to 8 significant digits. Just after rounding, the lower is reset to zero. The result is then normalized except in the case of operation code 02, *Unnormalized Floating Add Normalization.* Normalizing is the process of shifting the mantissa of the result to the left until a non-zero digit is in the high-order position. For each shift taken, a "1" is subtracted from the modified characteristic of the result.

32 FAD *(Floating Add).* This code causes the contents of the location specified by the data address to be added to the contents of the upper half of the accumulator. The sum is rounded, the lower is set to zero, and the sum is normalized.

02 UFA *(Unnormalized Floating Add).* This performs the same arithmetic function as code 32, *Floating Add,* except that the sum is not normalized.

33 FSB *(Floating Subtract).* This causes the contents of the location specified by the data address to be subtracted from the contents of the upper half of the accumulator. The difference is rounded, the lower is set to zero, and the difference is normalized.

37 FAM *(Floating Add Absolute).* This causes the absolute value of the contents of the location specified by the data address to be added to the contents of the upper half of the accumulator. The sum is rounded, the lower is set to zero, and the sum is normalized.

38 FSM *(Floating Subtract Absolute).* This causes the absolute value of the contents of the location specified by the data address to be subtracted from the contents of the upper half of the accumulator. The difference is rounded, the lower is set to zero, and the difference is normalized.

The execution of each of these five operations is similar and they are discussed together. In general, the arithmetic of each operation is accomplished this way:

When the contents of the location specified by the data address have been transferred to the distributor, the difference between the modified characteristics is found. If the modified characteristics are equal, the mantissas are ready to be added or subtracted. If the absolute value of the difference between the modified charactristics is greater than 8, the number with the smaller modified characteristic is ignored. If the absolute value of the difference between the modified characteristics is between 1 and 8, inclusive, the number with the larger modified characteristic is placed in the distributor, and the number with the smaller modified characteristic is placed in the upper half of the accumulator. The contents of the accumulator are then shifted right the number of places indicated by the difference between the modified characteristics. This shifting properly aligns the decimal point so that the mantissas can be added or subtracted. A detailed discussion of the operations follows.

1. The number in the location specified by the data address is transferred to the distributor. The operation code has already been examined to determine:
   a. Whether add or subtract
   b. Whether absolute or not absolute
   c. Whether normalize or unnormalize.
2. The modified characteristic of the number in the distributor is subtracted from the modified characteristic of the number in the upper half of the accumulator. Then,
   a. If the difference between the modified characteristics is zero, no decimal aligning is necessary. The mantissas are ready to be added or subtracted.
   b. If the difference between the modified characteristics is greater than 8, the modified characteristic of the number in the upper is restored. The contents of the upper are placed in the distributor and the upper is set to zero.
   c. If the difference between the modified characteristics is less than −8, the upper is set to zero.
   d. If the difference between the modified characteristics is between 1 and 8, inclusive, the modified characteristic of the number in the upper is restored, and the contents of the distributor and the upper are interchanged. Then, the contents of the accumulator are shifted to the right the number of places indicated by the difference between the modified characteristics.

e. If the difference between the modified characteristics is between −1 and −8, inclusive, the modified characteristic of the number in the upper is set to zero, and the contents of the accumulator are shifted to the right the number of places indicated by the difference between the modified characteristics.

3. The mantissa of the number in the distributor is added or subtracted to the upper half of the accumulator, positions 3 through 10. At the same time, the modified characteristic of the number in the distributor is inserted into the first and second positions of the upper.

4. If a mantissa overflow occurs during addition, the result is shifted right one position, a "1" is inserted in the high-order portion of the upper, and a "1" is added to the modified characteristic.

5. The result is rounded and the lower is set to zero. If an overflow occurs after rounding, a "1" is inserted in the high-order position of the upper and the modified characteristic is increased by 1.

6. The mantissa of the result is checked for zero. If zero, the modified characteristic is also set to zero. If the result is non-zero, the number is normalized unless operation code 02, *Unnormalized Floating Add,* is used.

The result is in the upper half of the accumulator, and the lower is zero. If the original two numbers have the same modified characteristics, the number originally brought to the distributor remains there. If the original two numbers have different modified characteristics, the number which has the larger modified characteristic is in the distributor.

The lower half of the accumulator should be zero at the start of an operation. However, the operation is completed, even though this condition is not met. Beginning an operation with a number other than zero in the lower creates erroneous digit positions to the right of the mantissa, and may result in incorrect mantissas and modified characteristics.

Examples of Operation Codes, 32, 02, 33, 37 and 38

			Contents of Drum Location	Accumulator Upper	Accumulator Lower	Distributor
32	FAD	Before	1000000050	9000000050	0000000000	1234567812
		After	1000000050	1000000051	0000000000	1000000050
32	FAD	Before	1212121250	3434343453	0000000000	1234567812
		After	1212121250	3435555553	0000000000	3434343453
32	FAD	Before	3434343453	1212121250	0000000000	1234567812
		After	3434343453	3435555553	0000000000	3434343453
32	FAD	Before	1234567860	1234567851	0000000000	1234567812
		After	1234567860	1234567860	0000000000	1234567860
32	FAD	Before	1234567851	1234567860	0000000000	1234567888
		After	1234567851	1234567860	0000000000	1234567860
32	FAD	Before	1000000099	9000000099	0000000000	1212121212
		After	1000000099	1000000000 (overflow)	0000000000	1000000099
32	FAD	Before	0000123457	0000000000	0000000000	1234123443
		After	0000123457	1234000053	0000000000	0000123457
			(non-normalized addition)			
32	FAD	Before	1212121250	3434343450	9000000000	1234567812
		After	1212121250	4646464750	0000000000	1212121250
			(lower contents not zero)			
32	FAD	Before	1212121251	1111111550	9000000000—	1234567887
		After	1212121251	1101010051	0000000000	1212121251
			(lower contents not zero)			
32	FAD	Before	1212121253—	3434343450	9999000000	1111111111
		After	1212121253—	1208686953	0000000000—	1212121253—
			(lower contents not zero)			
02	UFA	Before	1234567850—	1234569950	0000000000	8787939339
		After	1234567850—	0000002150	0000000000	1234567850—
33	FSB	Before	1234567850	7689432155	0000000000	0000000000
		After	1234567850	7689419855	0000000000	7689432155
37	FAM	Before	1234567850—	8800000050	0000000000	1231231234
		After	1234567850—	1003456851	0000000000	1234567850—
38	FSM	Before	1234567850	8800000050	0000000000—	1867543297
		After	1234567850	1003456851	0000000000—	1234567850

39 FMP *(Floating Multiply).* This causes the number in the location specified by the data address to be multiplied by the number in the upper half of the accumulator. The product after rounding and normalizing is in the upper, the lower is zero, and the multiplicand remains in the distributor.

Floating multiplication is accomplished by repeated addition. The number of additions is controlled this way:

1. The number in the location specified by the data address is transferred to the distributor.
2. The contents of the entire accumulator are shifted left one position. This shifting functions as previously discussed for all floating decimal operations. The shift left of one position places the high-order digit of the multiplier in a special storage position. This digit in the special position is then analyzed for zero or non-zero. If it is zero, another left shift is initiated. If it is non-zero, an add cycle is set up.
3. The mantissa of the multiplicand (in the distributor) is added into the lower half of the accumulator, positions 3 through 10. The digit in the special position is decreased by 1 each time the multiplicand is added. When the digit becomes zero, the addition operation is stopped and the contents of the accumulator are again shifted left one position to place the next digit of the multiplier in the special position.
4. After 8 such shift-and-add cycles, the two 8-digit mantissas have been multiplied together to produce a 16-digit product. The modified characteristic of the product is then placed in the 2 low-order positions of the upper. The modified characteristic of the product is the sum of the modified characteristics of the multiplier and the multiplicand, less 50.
5. If the high-order digit of the mantissa of the product is zero, the contents of the accumulator are shifted left one position and the modified characteristic is decreased by 1.
6. The mantissa of the product is rounded in the ninth decimal place, by adding 5 to, or subtracting 5 from, the high-order position of the lower. If an overflow is detected during the round cycle, the product must have had a mantissa 99999999 with a digit from 5 through 9 in the ninth place. When this type of overflow occurs, a "1" is inserted in the high-order position of the upper half of the accumulator and the modified characteristic is increased by 1. This results in a mantissa of 10000000 and a modified characteristic 1 greater than the one associated with the mantissa of 99999999.
7. Normalizing is now accomplished as a last function of the operation. If normal floating decimal numbers are used as factors, no normalizing is necessary.

The lower half of the accumulator should be zero at the start of the operation. Nevertheless, in most cases, the multiplication is completed if this condition is not met. However, the contents of the lower plays a part in developing the product. The absolute value of the contents of the lower are added to the absolute value of the high-order 10 positions of the product. This can produce an incorrect modified characteristic and an incorrect product. If the contents of the lower are sufficiently large, the multiplier can be altered by carrying into it.

Examples of Floating Multiply

		Contents of Drum Location	Accumulator Upper	Accumulator Lower	Distributor
39 FMP	Before	2222222251—	5555555550	0000000000	3765432085
	After	2222222251—	1234567951	0000000000—	2222222251—
39 FMP	Before	1234567850—	4111111157	0000000000—	8993246872—
	After	1234567850—	5075445456	0000000000	1234567850—
39 FMP	Before	5000000099	2000000053	0000000000	7681123890—
	After	5000000099	1000000002 (overflow)	0000000000	5000000099
39 FMP	Before	0000456752	0037657847	0000000000	1234567809
	After	0000456752	1720000043	0000000000	0000456752
		(non-normalized multiplication)			
39 FMP	Before	4567000048	3765780045	0000000000	1234567809
	After	4567000048	1719831743	0000000000	4567000048
39 FMP	Before	2000000050—	5555555550	2222222250	8765432017—
	After	2000000050—	3333333450	0000000000—	2000000050—
		(lower contents not zero)			
39 FMP	Before	1000000050—	5555555550	2222222250	1876932490—
	After	1000000050—	2777777850	0000000000—	1000000050—
		(lower contents not zero)			

34 FDV *(Floating Divide).* This causes the contents of the upper to be divided by the contents of the location specified by the data address. The quotient is moved to the upper and is rounded; the lower is set to zero, and the quotient is normalized.

There are no necessary conditions to be met when normalized floating-decimal numbers are used as the dividend and divisor. However, if non-normal numbers are used, the mantissa of the divisor must not have more leading zeros than the mantissa of the dividend. If this condition is not met, an unconditional overflow stop occurs. Division by zero also causes an unconditional overflow stop.

Floating-decimal division is accomplished by repeated subtraction. The number of subtractions is controlled this way:

1. The number in the location specified by the data address is transferred to the distributor.
2. The mantissa of the divisor is subtracted from the contents of the upper, positions 3 through 10. This subtraction continues until an overdraw occurs. When the overdraw occurs, the subtraction process stops, and the overdraw is corrected by adding the mantissa of the divisor to the upper. Following each subtraction in which an overdraw does not occur, a count of "1" is added to the second position of the lower.
3. Following the overdraw correction, the contents of the entire accumulator are shifted one position to the left and the process in step 2 is repeated. (This shifting functions as previously discussed for all floating-decimal operations.)
4. If the absolute value of the mantissa of the dividend is greater than or equal to the absolute value of the mantissa of the divisor, the repeated subtraction and shifting is continued until 9 quotient digits are developed. If the absolute value of the mantissa of the dividend is smaller than the absolute value of the mantissa of the divisor, the repeated subtraction and shifting is continued until 10 quotient digits are developed, the first of which is zero.
5. The modified characteristic of the quotient is now developed in positions 1 and 2 of the upper. If 9 quotient digits were developed, the modified characteristic of the quotient is equal to the modified characteristic of the dividend, minus the modified characteristic of the divisor, plus 51. If 10 quotient digits are developed, the modified characteristic of the quotient is equal to the modified characteristic of the dividend, minus the modified characteristic of the divisor, plus 50. Positions 3 through 10 of the upper are set to zero.
6. The contents of the entire accumulator are shifted left 8 positions as previously discussed for all floating decimal operations.
7. The quotient is rounded, the lower is set to zero, and the quotient is normalized.
8. The quotient is tested for zero. If zero, a zero modified characteristic is inserted.

The lower half of the accumulator must be zero at the start of the operation. Many unwanted situations can arise from a non-zero lower accumulator.

Examples of Floating Divide

		Contents of Drum Location	Accumulator Upper	Accumulator Lower	Distributor
34 FDV	Before	3450000053	5000000050	0000000000	8765433321
	After	3450000053	1449275448	0000000000	3450000053
34 FDV	Before	3450000053	1037667850	0000000000—	1987654321
	After	3450000053	3007732847	0000000000—	3450000053
34 FDV	Before	1760007513	1037667897	0000000000	0000000000
	After	1760007513	5895814634 (overflow)	0000000000	1760007513
34 FDV	Before	1760007575	1037667813	0000000000	1234567812
	After	1760007575	5895814688 (underflow)	0000000000	1760007575
34 FDV	Before	0001234550	0000543252	0000000000	2812818000
	After	0001234550	4400162052 (non-normalized division)	0000000000	0001234550

Floating-Fixed-Decimal Numbers and Unfloating-Floating-Decimal Numbers

It is possible to float fixed-decimal numbers and unfloat floating-decimal numbers by using two of the floating-decimal operations.

For example, an 8-digit field can be read into positions 3-10 of a word. A 5x can be emitted into the 2-low-order positions, (x is the digit represented by the number of places to the left of the decimal position in the field). Then, each of these numbers can be floating-added (code 32) to zero. The results will be normalized, and thus the modified characteristic is adjusted accordingly. If the numbers shown in the left-hand column below are entered with a 54 modified characteristic and floating-added to zero, the results are the numbers shown in the right-hand column.

0000.000154	1000000047
0000.001254	1200000048
0000.012354	1230000049
0000.123454	1234000050
0001.234554	1234500051
0012.345654	1234560052
0123.456754	1234567053
1234.567854	1234567854

As an example of unfloating, a group of floating-decimal numbers can be added using 02, *Unnormalized Floating Add,* to the improper decimal number, 00000000xx (xx is the largest modified characteristic in the group of floating-decimal numbers). If each of the numbers in the left-hand column is floating-added to 0000000056, the results are those shown in the right-hand column.

1234567849	0000000156
1234567850	0000001256
1234567851	0000012356
1234567852	0000123556
1234567853	0001234656
1234567854	0012345756
1234567855	0123456856
1234567856	1234567856

The numbers at the right could then be punched or printed in an 8-digit field, xxxxxx.xx.

Floating-Decimal Numbers in Conjunction with Other IBM 650 Operations

Floating-decimal numbers can be operated on, using basic 650 operations. In this case they are interpreted in the same way as any other 10-digit number. For example, 0000000001 can be added to 1234567852 to produce 1234567853. This operation, by adding "1" to the modified characteristic, has the effect of multiplying the floating-decimal number by 10. It is possible to test the floating-decimal number to determine whether it is zero or non-zero, positive or negative. By shifting, the modified characteristic can be separated from the mantissa, and either can then be modified by programming. The facility for operating on numbers with either type of arithmetic operation provides great flexibility.

The data address of a floating-decimal instruction may be 8005, 8006, or 8007, in which case the contents of the specified indexing register are used in the operation as a non-normalized floating-decimal number. For example, if an indexing register contains 1051, this number could be used as 0000001051 in any floating-decimal operation, 51 being the modified characteristic.

Optimizing Floating-Decimal Operations

Because the total of word-times for a floating-decimal operation is a function of both factors used in the calculation, definite rules cannot be stated for optimizing the data address of an instruction following a floating-decimal operation. Figure 18 gives the optimum location for the data and instruction addresses for each of the floating-decimal operations. Also shown is the maximum number of word-times for any floating-decimal operation, when both factors are known.

Figure 19 is a sequence chart for 650 floating-decimal operations.

Op Code	d=n+		i=d+		Maximum Time Normalized		Maximum Time Non-Normalized	
	n Even	n Odd	d Even	d Odd	d Even	d Odd	d Even	d Odd
32	3	3	5	4	48	47	62	61
33	3	3	5	4	48	47	62	61
34	3	3	5	4	244	243	244	243
37	3	3	5	4	48	47	62	61
38	3	3	5	4	48	47	62	61
39	3	3	5	4	174	173	200	199
02	3	3	5	4	36	35	36	35

Figure 18. Optimizing Floating-Decimal Operation

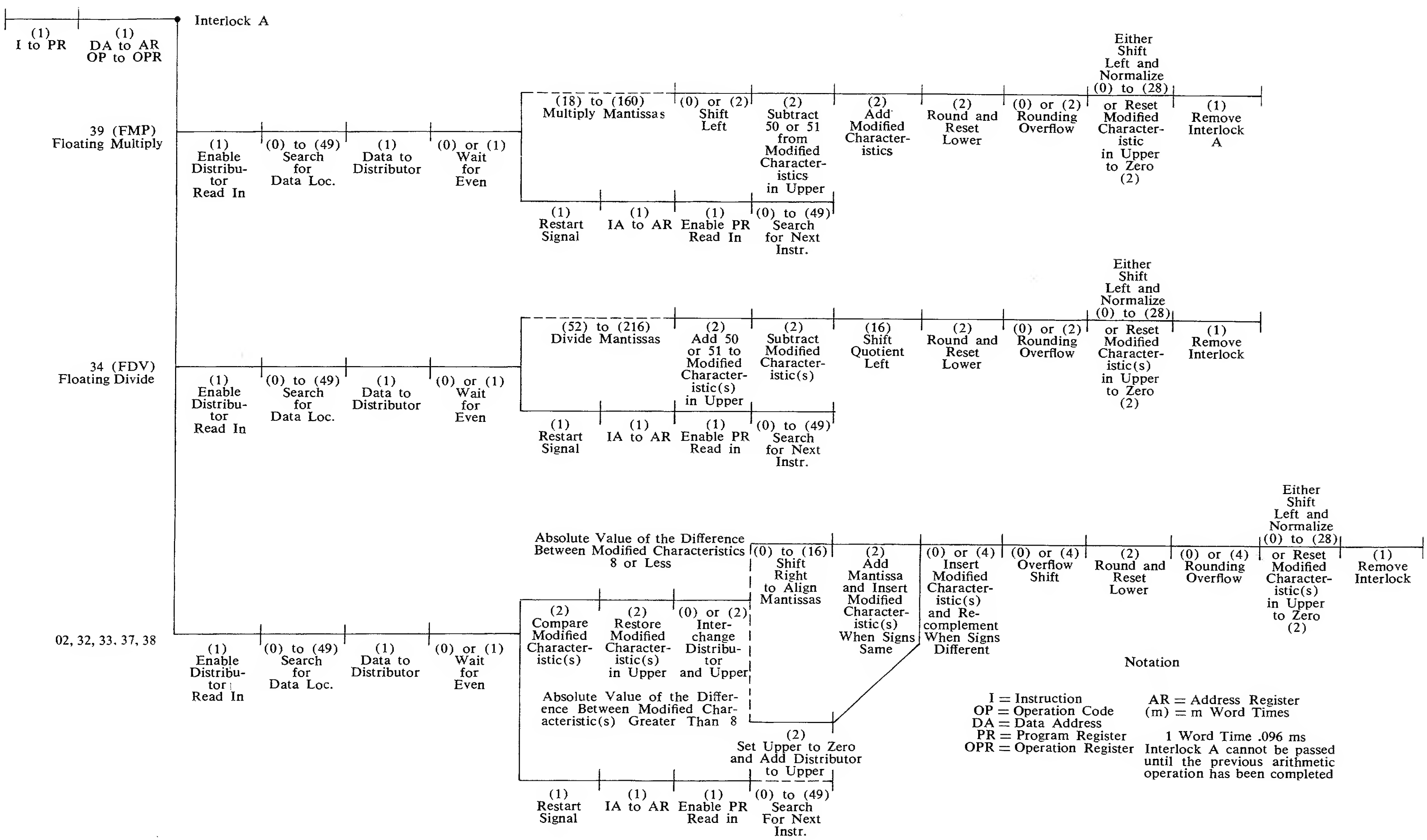


Figure 19. Sequence Chart for 650 Floating Decimal Operations

Magnetic Tape

Magnetic tape provides business with a storage medium in which large volumes of data can be written on a few reels of tape. Information is recorded in the form of magnetized spots. Magnetic tape can be used for repetitive processing. Information is automatically erased before new information is written. The lightweight, compact, and durable quality of the tape provides ideal permanent record storage. Figure 20 is a schematic of a 650 system with tapes.

IBM 727 Magnetic Tape Unit

The IBM 727 Magnetic Tape Unit (Figure 21) is used with the IBM 650 Data Processsing System. It is the same unit that is used with the IBM 705. The tape character coding is also the same. There is a difference in record format. The 650 reads or writes a tape record of one word up to a maximum of 60 words for each record.

When tape units are used with a 650 system, an IBM 652 Control Unit and an IBM 653 Storage Unit with 60 words of immediate access storage are required. Ten additional operation codes are available to the programmer with a 650 tape system.

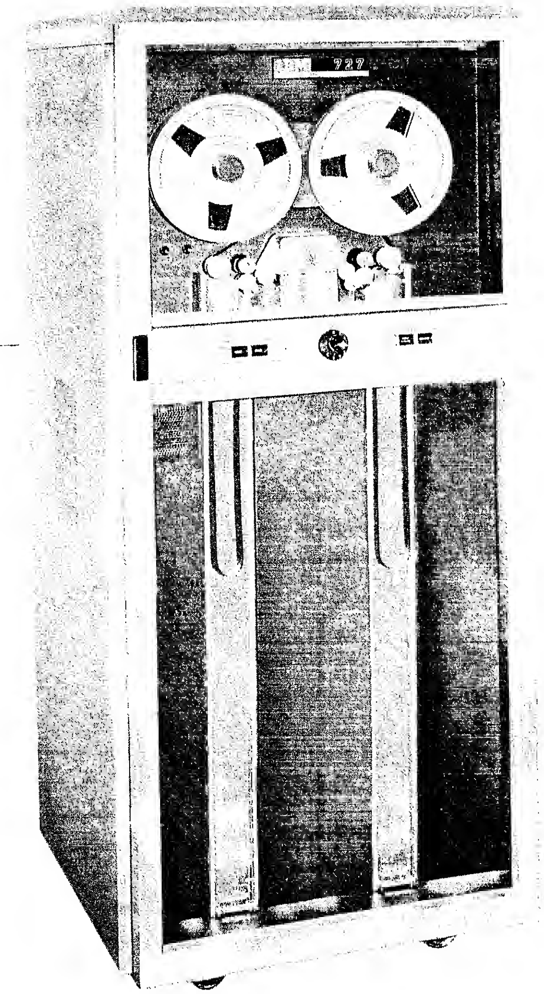


Figure 21. IBM 727 Magnetic Tape Unit

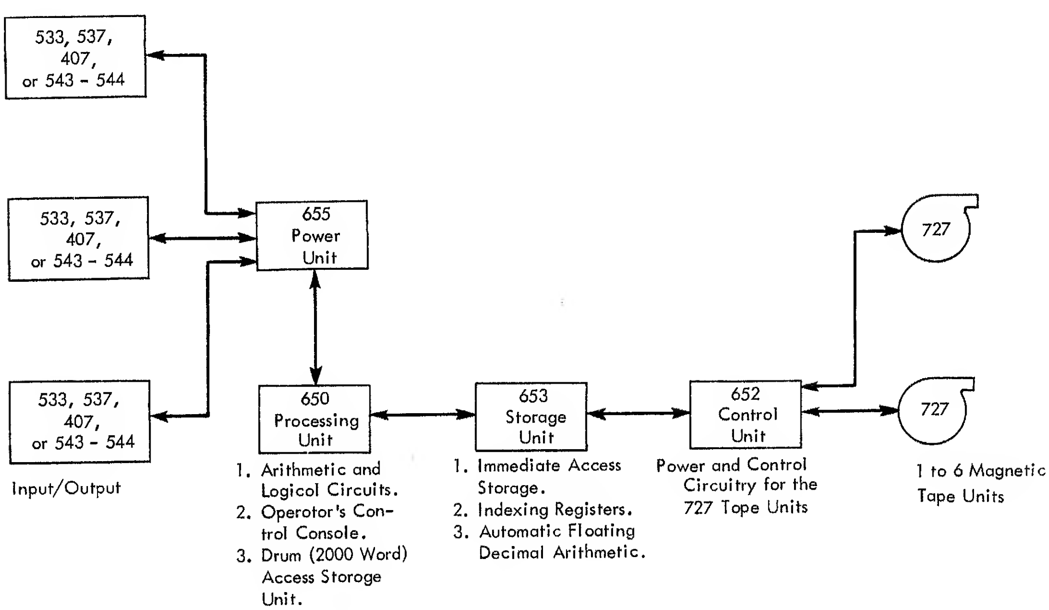


Figure 20. Schematic of 650 System

IBM 727 Specifications

Tape specifications for the IBM 650 Data Processing System:

1. Tape density is 200 characters per inch.
2. Tape reading, writing and backspacing speed is 75 inches per second.
3. Tape reels are 10½ inches in diameter with a capacity of 2400 feet of magnetic tape.
4. Each reel is equipped with a removable File-Protection Ring. If this ring is not in place, writing on tape is not possible.
5. Inter-record gap is approximately ¾ inch. The inter-record gap (IRG) is due to acceleration and deceleration time when a tape unit is used: approximately ⅜ inch to accelerate, and ⅜ inch to decelerate.

IBM 727 Used with a 650 System

1. Record size is from 1 to 60 words. A tape record is defined as the information between two consecutive inter-record gaps.
2. All tape records are written from, or read into, immediate access storage. Therefore, simultaneous tape read-write is not possible for a standard tape system.
3. Up to six IBM 727 Magnetic Tape Units can be attached to the IBM 650 System. These units are capable of reading or writing tape. They can be used as input or output in any combination, such as three units for input and three units for output. To permit this type of operation, each tape unit is assigned an address which can be selected through programming. The tape unit addresses are:

8010	8013
8011	8014
8012	8015

The units position of each address (0 through 5) is set on a dial on the front of each tape unit. Therefore any tape unit can fulfill the address requirements of any program. Each unit can have any of the six numbers because no unit has a preassigned number. Two units should not have the same address at the same time.

Character Coding on Tape

All information represented on 650 magnetic tape is recorded in a seven-channel code by the IBM 727 Magnetic Tape Unit. This is the same code that is used by the IBM 705 Data Processing System. It is often referred to as BCD (Binary Coded Decimal).

The seven channels, or tracks, are divided into three sections on the tape:

1. The numerical section — four channels with channel assignments 1, 2, 4, and 8.
2. The zone section — two channels called A and B.
3. The checking section — one channel called C.

The arrangement of magnetic bits in different combinations in these seven channels is used to represent digits, alphabetic characters, and special characters. Digits are represented by placing bits in the appropriate numerical channels. Alphabetic and special characters are represented by placing bits in the appropriate numerical and/or zone channels. A magnetic bit is placed in the checking channel whenever the total count of the bits, for any one character, in the numerical and zone channels is odd. Thus every character is represented by an even number of bits in the vertical direction. Figure 22 is a schematic representation of digits, alphabetic characters, special characters, blanks, and a tape mark.

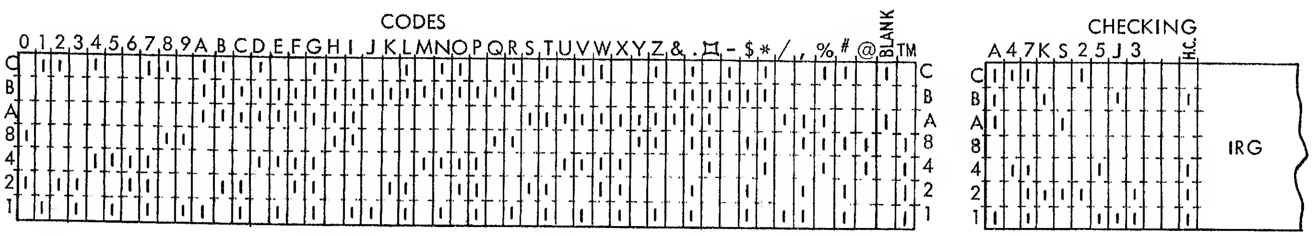


Figure 22. Schematic of Tape Codes

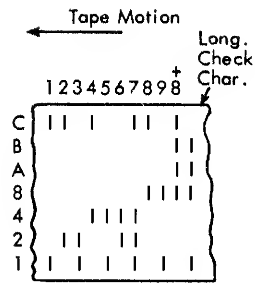


Figure 23. One-Word Tape Record with Longitudinal Check

Checking

Information written on tape or read from tape is checked to insure accuracy of data recording and data transmission.

When the tape unit writes a record, an odd-even count is made horizontally on each channel. At the end of the record a longitudinal check character is written. This character is such that it causes the total number of bits in each channel to be an even number. Figure 23 is a schematic representation of a one-word tape record with its longitudinal check.

The vertical check bit and the longitudinal check character are recorded on tape during the write operation, to provide a means of determining that the record is read correctly on any subsequent read operation. If, when reading, any character fails to pass the vertical redundancy check, or any bit channel fails to pass the longitudinal redundancy check, an error condition is indicated for the record being read. Checking is also provided to insure that the information is recorded correctly. In the write operation, the impulses sent to the recording unit for each character are analyzed (echo check) for an even bit-count to insure that a valid character is at the recording unit. Through programming, the machine is stopped when any irregularity is encountered in either tape reading or writing.

Photo-Sensing Markers

Photo-sensing markers, also referred to as *reflective spots,* are placed on the tape to enable the tape unit to sense where reading and writing are to begin and to stop. The markers are small pieces of plastic, one inch by 3/16 inch, coated with vaporized aluminum on one side and with adhesive on the other. They are fastened to the base (uncoated) side of the tape. The photo-electric cells sense them as either the load point marker where reading or writing is to begin on tape or as the end-of-reel marker where writing is to stop.

Approximately 18 feet of tape should be allowed between the beginning of the reel and the load point marker as a leader for threading the tape over the feed rolls and the read-write head. Information cannot be stored in this space. To indicate the load point, the one-inch dimension of the marker must be parallel to, but not more than 1/32 inch from the channel 1 edge of the tape — the edge nearest the operator when the reel is mounted (Figure 24).

Approximately 18 feet of tape are reserved between the end-of-file marker and the physical end of the tape attached to the hub of the machine reel. To indicate end of file, the marker must be placed parallel to, but no more than 1/32 inch from, the C track edge of the tape (the edge nearest the tape unit when the reel is mounted, Figure 24). Sensing this spot during a write operation causes the end-of-file indicator to be turned on.

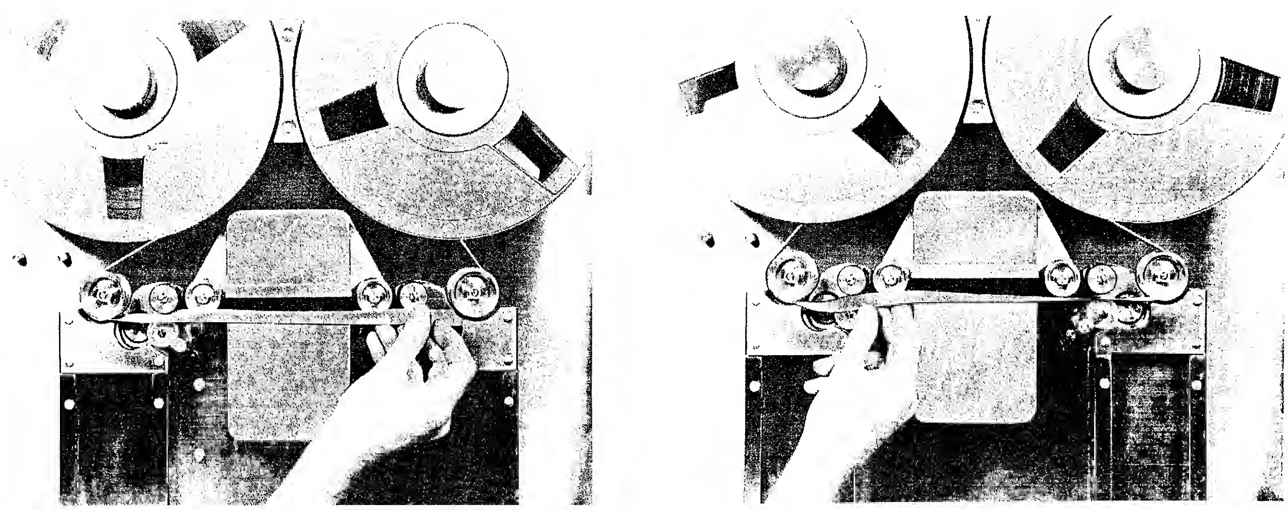


Figure 24. Reflective Spots on Tape

Place load point and end-of-file markers on tape with care. They should be properly aligned and pressed tightly onto the tape with the back of the fingernail.

It is best to do this while the tape is loaded on a unit, to reduce the collection of dust on the unrolled tape. If this is done away from the unit, keep the unrolled end of tape off the floor and away from dusty areas.

File Protection

The back of the tape reel (machine side) has a circular groove in which a plastic ring may or may not be inserted. To write on a tape, the plastic ring (file protection ring) must be placed in the groove of the tape reel (Figure 25A). A tape can be read with the file protection ring inserted or removed. The file protection ring should be removed from the tape reel after writing on tape is completed (Figure 25B). Doing this prevents accidental writing and resultant loss of valuable tape records. Never remove the file protection ring while tape is loaded in the vacuum columns of the tape unit. Doing this could cause a broken or damaged tape.

Reading and Writing Tape

The 650 tape system provides two modes of reading and writing tapes: the numerical mode and the alphabetic or alphanumerical mode.

Numerical Mode

In the numerical mode, records consisting of from one to sixty words of numerical data are read from tape into immediate access storage or written from storage onto tape. Transfer of data is on a one-to-one basis: one digit in storage becomes one character (in 705 code) on the tape, and vice versa. Records of this type are read by means of operation code 04, RTN (Read Tape Numerical) and are written by using operation code 06, WTN (Write Tape Numerical).

The sign of a numerical word is represented on tape by appropriate recorded bits in both the A and B channels; a minus sign is recorded as a bit in the B channel. Thus the units position of each numerical word on tape is actually an alphabetic or special character. Figure 26 shows a two-word numerical tape record as it appears in immediate access storage and on tape.

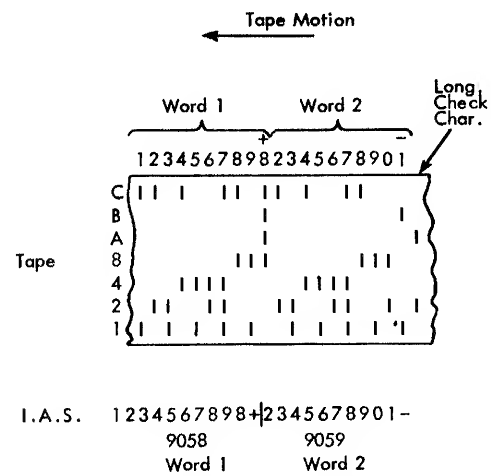


Figure 26. Two-Word Tape Record in IAS — Tape

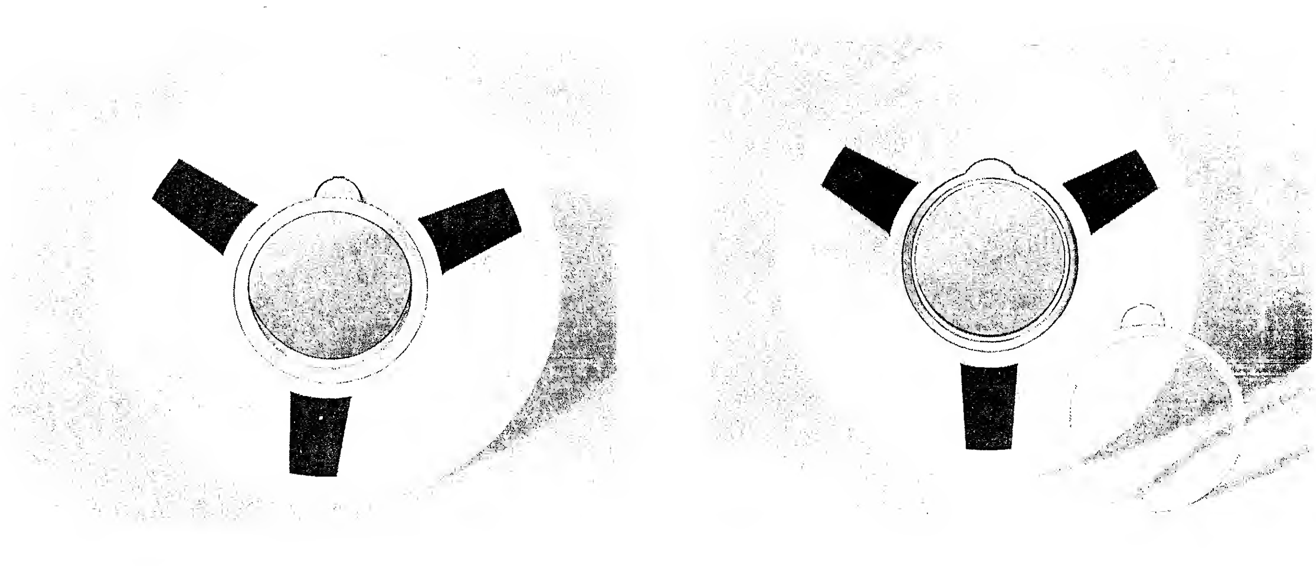


Figure 25. Tape Reel — File Protection Ring

Inasmuch as an alphabetic character is represented within the 650 by two decimal digits A = 61; I = 69; J =71; R = 79; etc., alphabetic and alphanumerical information can be treated as numerical data and written on, or read from, tape in numerical form. Since, in this case, the information is recorded on tape in the 650 double-digit representation as numerical data, this procedure is feasible only if the tape is to be used for further processing in the 650, and not in peripheral equipment. Figure 27 is a schematic representation of a three-word alphanumerical record in numerical form, in immediate access storage and on tape.

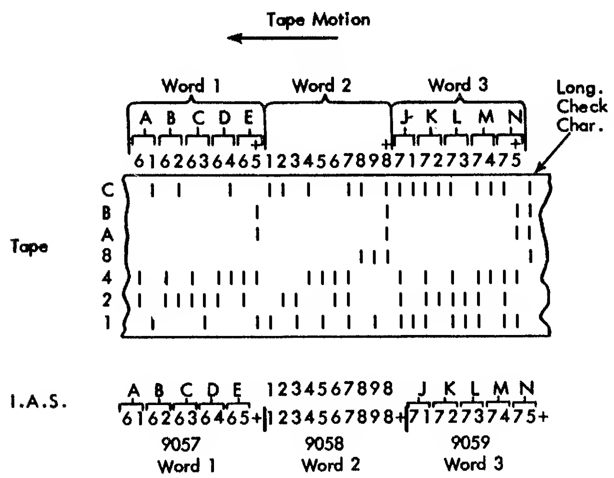


Figure 27. Three-Word Alphanumerical Record

TAPE READING NUMERICAL

Records read from a tape are automatically entered into the immediate access storage unit. The size of the record, variable from 1 word to 60 words, must be known so the immediate access storage timing ring can be set to accept a record of that size. The record from the tape must exactly fill immediate access storage from the predetermined starting point (setting of timing ring) through word 9059. If the tape record is either too short or too long, an error is indicated and the machine automatically stops.

04 RTN *(Read Tape, Numerical).* A numerical record is read from the tape unit specified by the data address. This record is transferred into immediate access storage, and is checked for valid character coding, and over-all record validity. The tape record is entered into immediate access storage beginning with a predetermined word (setting of timing ring) and continues entering information until location 9059 has been entered or the end of record gap is encountered. If the end of record gap and the end of immediate access storage are not encountered at the same time, an error is signalled.

TAPE WRITING NUMERICAL

All tape records are written from the immediate access storage unit. The size of the records may vary from 1 to 60 words. This size is determined by specifying the point in immediate access storage from which writing is to begin (setting of timing ring). Writing continues from the starting point through word 9059 (end of immediate access storage).

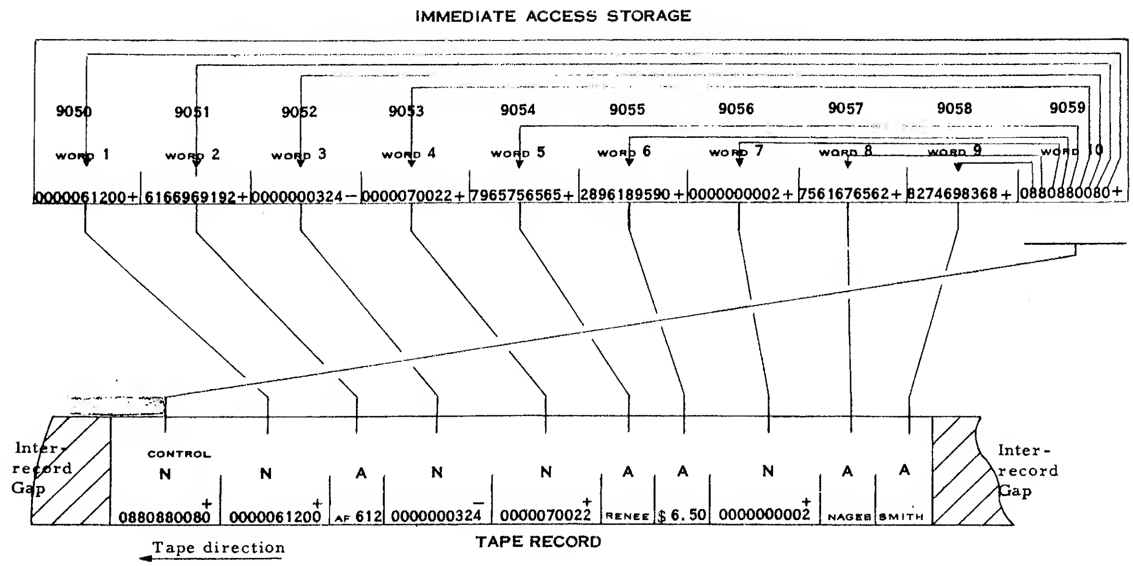
06 WTN *(Write Tape, Numerical).* A numerical record is written from immediate access storage onto the tape unit specified by the data address. The record is written beginning with a specific location (setting of the timing ring) in immediate access storage, and continues writing through location 9059.

Alphabetic or Alphanumerical Mode

Alphabetic or alphanumerical records are read or written in multiples of ten words, using operation codes 05 (RTA — Read Tape Alphanumerical), and 07 (WTA — Write Tape Alphanumerical). Tapes written in this mode can be processed in IBM 700 series equipment. A record may consist of from 1 to 6 such blocks of ten words within each block.

The ten-word block, the basic unit of the alphanumerical record, consists of nine words of data and one control word indicating which words of the block are numerical and which are alphanumerical. Figure 28 is a schematic representation of a one-block alphanumerical record in immediate access storage. Figure 28 also shows the same ten words as they would appear on tape.

This automatic conversion between 650 two-digit representation, and single character representation takes place in the 652 control unit. This method of writing tape records requires that the alphanumerical record be composed of 10-word blocks. The conversion between two-digit and single character representation is determined by an alpha control word. One alpha control word is needed for each 10-word block. The alpha control word specifies which of the other 9 words of the same block are alphanumerical and which are numerical. In this manner, alphabetic data (two-digit representation) is converted to single character 705 code, and numerical data (single-digit representation) is kept in single-digit form.



N signifies a numerical word.
A signifies an alphanumerical word.
9050—9059 are the I A S addresses of the words.

Figure 28. One Block Alphanumerical Record in IAS and on Tape

Before an alpha-mode tape command (05, 07) is executed, the timing ring must be set to the first word of a block in IAS (9000, 9010, 9020, 9030, 9040, 9050). If the timing ring is not set at the beginning of a block, erroneous tape reading or writing can occur. It is possible that this type of error can be processed without causing an error signal.

Alphabetic Control Word

The alphabetic control word for each block indicates which of the nine preceding words are alphanumerical or numeric. The control word for each block is always placed in the last word of the block 9009, 9019, 9029, 9039, 9049, and 9059). The units position of the control word determines whether the first word of the block is numerical or alphanumerical. The 10's digit of the control word determines whether the second word of the block is numerical or alphanumerical, etc. The high-order position of the control word is unused (Figure 28). An *8* in a digit position indicates that the corresponding word is to be alphanumerical. Any digit other than an *8* indicates that the corresponding word is numerical. The high-order digit may contain any value. If the corresponding word is to be numerical, it may be desirable to place a *9* in the corresponding position of the control word. This will enable testing the control word in the program by use of the branch distributor operation codes.

07, WTA *(Write Tape Alphanumerical).* An alphanumerical record is written from immediate access storage into the tapes specified by the data address. Writing must take place beginning with word 9000, 9010, 9020, 9030, 9040, 9050. This is determined by the setting of the timing ring. When this operation code is used, the tenth word, and every tenth word following, is the control word for the nine words that precede it. Because alphabetic and special characters require two positions of storage, a maximum of 45 alphanumerical characters, plus 10 numerical characters in the control word, can be written for each ten words of immediate access storage. Writing continues from the specified starting point through 9059.

05, RTA *(Read Tape Alphanumerical).* An alphanumerical record is read from the tape unit specified by the data address. This record is read into immediate access storage and is checked for valid character coding and over-all record validity. All alphanumerical records must be multiples of 10 words in length (10, 20, 30, etc.). The first word of each group of 10 is the control word for that group, and it specifies which of the following words are alphabetic and which are numerical. These control words are read into words 9009, 9019, 9029, etc.

Each alphabetic character is stored in immediate access storage as a 2-digit number. Therefore, in each group of 10 words, there may be a maximum of 45 alphabetic characters in addition to the control word. A 60-word record from the tape fills immediate access storage by placing the first alphabetic control word in location 9009 and the following 9 words from the tape in location 9000 through 9008. The second alpha-control word is placed in location 9019 and the following 9 words are placed in locations 9010 through 9018, etc.

Selection of Tape Read/Write Mode

If tape prepared by the 650 is to be used only as input to the 650 system, select the read/write mode which allows the maximum number of records to be written on a reel of tape. This results in economic use of tape reels and a reduction of read/write time. Consideration should also be given to the grouping of records and processing for alphabetic control.

When a record contains alphabetic information it may not always be advantageous to write in alpha-mode.

EXAMPLE 1. A master record contains 20 alphabetic characters and 50 digits.

Alpha Mode		Numerical Mode
10	Digits in control word	0
20	Alpha character representation	40
50	Numeric digits	50
80	Total characters on tape	90
25,043	Records per reel	24,000

The alpha-mode should be selected.

EXAMPLE 2. A master record contains 10 alphabetic characters and 40 digits.

10	Digits in control word	0
10	Alpha character representation	20
40	Numeric digits	40
15	Alpha zeros to complete the 10 word block	0
75	Total characters on tape	60
25,600	Records per reel	27,428

The numeric-mode should be used.

EXAMPLE 3. A master record that contains 25 alphabetical characters and 20 digits results in:

a. an alpha-mode tape record of 65 characters (26,790 records per reel)

b. a numeric-mode tape record of 70 characters (26,181 records per reel)

But the maximum record grouping of an alpha record is 6, and the maximum grouping of the numerical mode record in this example is 8 and this results in:

a. the alpha-mode record — 63,996 records per reel

b. the numeric-mode record — 64,896 records per reel

A 650 tape record can be composed of any multiple of 10 tape characters up to 50, and multiples of 5 from that point on, except for 105. The total cannot exceed 600 characters.

Figure 29 shows the relationship between characters per record, inches per record, and records per reel. In this chart inter-record gaps are considered in determining the number of inches allotted to a record.

Char./ Record	Inches/ Record	Records/ Reel	Char./ Record	Inches/ Record	Records/ Reel
10	.800	36,000	320	2.350	12,255
20	.850	33,882	325	2.375	12,126
30	.900	32,000	330	2.400	12,000
40	.950	30,315	335	2.425	11,876
50	1.000	28,800	340	2.450	11,755
55	1.025	28,097	345	2.475	11,636
60	1.050	27,428	350	2.500	11,520
65	1.075	26,790	355	2.525	11,405
70	1.100	26,181	360	2.550	11,294
75	1.125	25,600	365	2.575	11,184
80	1.150	25,043	370	2.600	11,076
85	1.175	24,510	375	2.625	10,971
90	1.200	24,000	380	2.650	10,867
95	1.225	23,510	385	2.675	10,766
100	1.250	23,040	390	2.700	10,666
110	1.300	22,153	395	2.725	10,568
115	1.325	21,735	400	2.750	10,472
120	1.350	21,333	405	2.775	10,378
125	1.375	20,945	410	2.800	10,285
130	1.400	20,571	415	2.825	10,194
135	1.425	20,210	420	2.850	10,105
140	1.450	19,862	425	2.875	10,017
145	1.475	19,525	430	2.900	9,931
150	1.500	19,200	435	2.925	9,846
155	1.525	18,885	440	2.950	9,762
160	1.550	18,580	445	2.975	9,680
165	1.575	18,285	450	3.000	9,600
170	1.600	18,000	455	3.025	9,520
175	1.625	17,723	460	3.050	9,442
180	1.650	17,454	465	3.075	9,365
185	1.675	17,194	470	3.100	9,290
190	1.700	16,941	475	3.125	9,216
195	1.725	16,695	480	3.150	9,142
200	1.750	16,457	485	3.175	9,070
205	1.775	16,225	490	3.200	9,000
210	1.800	16,000	495	3.225	8,930
215	1.825	15,780	500	3.250	8,861
220	1.850	15,567	505	3.275	8,793
225	1.875	15,360	510	3.300	8,727
230	1.900	15,157	515	3.325	8,661
235	1.925	14,961	520	3.350	8,597
240	1.950	14,769	525	3.375	8,533
245	1.975	14,582	530	3.400	8,470
250	2.000	14,400	535	3.425	8,408
255	2.025	14,222	540	3.450	8,347
260	2.050	14,048	545	3.475	8,287
265	2.075	13,879	550	3.500	8,228
270	2.100	13,714	555	3.525	8,170
275	2.125	13,552	560	3.550	8,112
280	2.150	13,395	565	3.575	8,055
285	2.175	13,241	570	3.600	8,000
290	2.200	13,090	575	3.625	7,944
295	2.225	12,943	580	3.650	7,890
300	2.250	12,800	585	3.675	7,836
305	2.275	12,659	590	3.700	7,783
310	2.300	12,521	595	3.725	7,731
315	2.325	12,387	600	3.750	7,680

Figure 29. Number of Records on a Reel of Tape

When a record contains only 10 characters, a total of 36,000 records can be written on a reel of tape. If a record contains 600 characters, only 7,680 records can be written on a reel. This table is useful in determining the number of records that can be written on a reel of tape; i.e., if your record contains 210 characters, 16,000 records can be written on a reel of tape.

Miscellaneous Tape Operating Codes

03 RTC *(Read Tape Check).* A record is read at the tape unit specified by the data address. This record is not transferred into immediate access storage but is checked for character validity only. Immediate access storage is free during the time in which this record is read. For example, if an IAS address is encountered in the program during tape read time, the 650 program is not interlocked. If an RTC is to be effective, it must be followed by a 25 NTS and a 54 NEF operation code.

55 RWD (Rewind Tape). The tape in the unit specified by the data address is rewound. At the completion of the rewind operation, the tape is at the load-point and any tape operation can now be performed on this unit. During the rewind operation, (approximately 1.2 minutes for a full reel) another tape unit can operate concurrently.

56 WTM *(Write Tape-Mark).* A tape-mark is written on the tape unit specified by the data address. The tape-mark is normally written after the last record has been placed on a reel of tape. When this tape is processed as input, the tape-mark turns on the end-of-file indicator and tape-signal indicator for testing by the 25 NTS and 54 NEF operation codes.

57 BST *(Backspace Tape).* The tape in the unit specified by the data address is backspaced one record. The reading of a tape record and backspacing the tape to write a new record over the record just read should never be attempted. When the new record is being written, the old record is erased, and this erasure may continue into and partially destroy the next record of the file.

Tape Error and End-of-File Conditions

To detect the presence of a tape error or an end-of-file condition the tape-signal indicator and the end-of-file indicator are interrogated. These indicators should be tested by the program after each tape read or write operation. Failure to do this may result in processing erroneous information because the indicators are always reset by the next tape command, which refers to an 801X address. If an end-of-file and an error condition occur on the same read or write operation, the end-of-file condition is ignored.

The tape-signal indicator is ON whenever a tape error or end-of-file condition is encountered. When an end-of-file condition occurs, the tape indicate light on the selected tape unit is ON. The selected tape unit is the last tape unit addressed. The select light is ON on this unit until the next valid 801X data address is encountered in the 650 program.

Tape errors include: invalid characters, incorrect longitudinal check character, a long record, a short record, or a failure in core storage during a tape read or write operation.

The end-of-file indicator and the tape-signal indicator are ON whenever an end-of-file condition (tape mark while reading tape, or the reflective spot while writing tape) is encountered.

25 NTS *(Branch No Tape Signal).* Operation code 25 NTS tests the condition of the tape-signal indicator. If either an end-of-file or error condition exists, the tape-signal indicator is ON, and the next instruction is taken from the I-address. If neither an end-of-file nor an error condition exists, the tape signal indicator is OFF, and the next instruction is taken from the D-address. Thus, the normal routine is programmed from the D-address the end-of-file and tape error routines from the I-address. This operation code does not turn off the tape-signal indicator which it interrogates. When the next valid 801X data address is encountered, the tape-signal indicator is turned off.

54 NEF *(Branch No End of File).* This operation code tests the condition of the end-of-file indicator. It should always be preceded by a 25 NTS operation code. If the end-of-file indicator is OFF (indicating a tape error condition), the next instruction is taken from the D-address. If the end-of-file indicator is ON (indicating an end-of-file condition), the next instruction is taken from the I-address. The 54 NEF operation code does not turn off either the end-of-file indicator or the tape-signal indicator, but it does turn off the tape indicate light on the selected tape unit.

Programming for NTS and NEF

Each time a tape read or write operation is performed this procedure should be followed.

STEP 1. Test for tape signal (Op code 25 NTS)
- If no tape signal, take D-address (continue program).
- If tape signal indicator is on, take I-address and step 2.

STEP 2. Test end-of-file (Op code 54 NEF)

If not end-of-file, take D-address, which indicates a tape error (error routine).

If end-of-file, take I-address (end-of-file routine).

The tape-signal indicator and end-of-file indicator are reset by the next tape instruction. Therefore, it is important that each time a tape record is read or written the tape-signal and end-of-file indicators be tested. If a tape record is read or written, and these indicators are not tested before the next tape instruction is encountered, any error or end-of-file condition is lost.

LOCATION OF INSTRUCTION	INSTRUCTION OP	DATA	INSTRUCTION	OPERATION ABBRV.
1829	54	Tape Error	1830	NEF
1830	56	8012	1840	WTM
1840	55	8012	1900	RWD
1900	66	1803	1907	RSL
1907	20	1803	1908	STL
1908	15	0020	1925	ALO
1925	20	0020	1923	STL
1923	69	1830	1901	LDD
1901	22	1830	1933	SDA
1933	69	1840	1902	LDD
1902	22	1840	0025	SDA
Return to Previous Routine.				

END-OF-FILE ROUTINE

The end-of-file routine for an output tape consists of writing the tape mark and programing a rewind operation. (The tape mark is used for the end-of-file detection when the tape is used for input.) Following the rewind instruction, the end-of-file routine may take different forms depending upon the number of tape units used for output. When only one tape unit is used, the program is normally written to stop the 650, allowing the operator to change tapes. If more than one tape unit is used, the end-of-file routine is normally written to alter the address of the output tape unit in the program. This allows the program to continue processing data, writing the output on the second or alternate tape unit. The operator can then change the tape in the first unit in the conventional manner without stopping the 650. This results in more efficient use of machine time.

The end-of-file routine for input tapes is handled the same way as for output tapes. When one tape unit is used, the machine is stopped to change tapes. With two tape units in use, the addresses in the program can be altered to permit processing to continue to the second tape while the operator changes the first. The only difference is establishing an end-of-file condition. The reflective spot indicates approximately where the physical end of the tape is reached, and is not exact enough to indicate where the last record on the tape has been written. That's why a tape mark is recorded after the last actual tape record. When read, this tape mark turns on the tape-signal and end-of-file indicators. Instructions then select the end-of-file routine for the input tape. The following program is an end-of-file routine using two output tapes:

TAPE ERROR ROUTINE

All tape read or write routines should contain a subroutine to correct reading or writing errors. Tape can be backspaced (BST 57) and either rewritten or reread. A count routine can be set up to limit the number of times this is done. Read error routines usually provide for reading the tape record causing the error, three times before stopping the program. The program can be planned to allow the operator to try reading the tape record three more times by pressing the program start key.

Write error routines usually provide for a single additional attempt at rewriting the record before the program is stopped. This routine can also be planned to attempt rewriting by pressing the program start key.

Read or write error routines eliminate a large part of the need for manual intervention on tape error conditions caused by invalid characters in reading or writing. It does not correct errors in program logic, such as improper setting of the timing ring.

NOTE: Do not attempt to insert a record into a tape that contains active records. Reading a tape record and backspacing the tape to write a new record over the record just read should never be attempted. When the new record is written, the old record is erased. This erasure may continue into and partially destroy the next record of the file.

TAPE ERROR PUNCH OR PRINT OUT

If a tape error cannot be corrected by rereading or rewriting the record, the contents of core storage can be punched or printed out. This is particularly valuable in a tape read operation.

Execution of an output instruction (71, 74, 77) with a valid 90XX data address, and with the tape error indicator on, sets up special conditions:

1. The validity check at the output of core storage is by-passed (permitting the transfer of invalid information).
2. An early impulse is emitted from the MTC (magnetic tape check) hub on the output unit control-panel.

MTC *Impulse.* This MTC (Magnetic Tape Checking) impulse can be wired to PVC to by-pass output validity checking; it can be used to pick up selectors for format control, or to print or punch error identifications.

The timing of this impulse on the IBM 533, will enable the selection of control information, punching format, the input/output checking hub, and DPBC on a 2-cycle delay basis. In the IBM 407 the MTC impulse may be wired to the pickup of a co-selector or pilot selector. Control information in the 407 can only be selected through the co-selector points. However, the 407 pilot selectors transfer in time to select all-cycle impulses.

As long as the tape error indicator is on, any output command (71, 74, 77) with a valid 90XX data address causes an MTC impulse.

A vertical redundancy error causes all bits to be filled in the corresponding position of core storage. If this character is punched, it laces a card column; if printed, a % symbol prints. The other types of errors may not be as readily detected because they can cause a multiple punched column, a blank column or printing position, or some character other than the correct one on the printer.

The tape error condition can be reset only by a succeeding tape command, or by depression of the error sense reset key. The HLT and NOP operations, with a data address of 801X, also reset the tape error condition. If tape errors are to be written out, the programmer should avoid resetting the error condition until all the necessary write instructions are given. Otherwise, the MTC impulse does not emit.

NOTE: Under certain conditions (print or punch error records, and their corresponding master tape or transaction records) difficulties may be encountered in the format of the printed or punched records. This difficulty arises from the operation of the MTC impulse. The MTC hub emits an early impulse for the selection of a special error record format whenever the tape error condition is set and a write 90XX instruction is given.

This means that if both the error record and the transaction record are written from IAS and no tape operations are executed between the execution of the two write instructions, both the error and the transaction record would be printed in the same format, and any emitted error indication would appear for both records.

To avoid printing both records in the error format, a program which includes an error punch or print-out should provide for writing the error record from IAS and the correct record from the drum. If both records are to be written from IAS, the error record should be printed first and some tape operation (such as BSP) should be executed before the correct record is printed, in order to reset the tape error, and to prevent the MTC hub from emitting an impulse when the correct record is printed.

Timing Considerations

One of the objectives of programming is to use the efficiency that was designed into the IBM 650. The philosophy of 650 design permits overlapping many operations:

1. Input/output synchronizers allow processing to overlap input/output operations.
2. The internal flow paths allow processing to overlap the search for the next instruction.
3. The two-address instruction format permits optimum programming.
4. The core storage unit (IAS) permits processing to overlap tape read and write operations, provided the processing program does not refer to IAS.
5. The tape interlocks, *Tape Control Interlock* (TC-I) and *Tape Unit Interlock* (TU-I), permits certain tape functions to progress concurrently.

Interlocks

Immediate Access Storage Interlock (IAS-I). This interlock is set whenever an instruction requires the use of IAS for its execution. It is released when the operation is completed. On an operation that can be overlapped, and that also requires a relatively long time to complete, this interlock prevents the interruption of the first operation by subsequent program instructions referring to IAS.

Thus, the IAS interlock is set at the beginning of any tape read-write operation (Op codes — 04, 05, 06, 07), and is removed when the tape operation is completed. While the tape operation is in progress, program execution can continue as long as no reference is made to IAS. If the program tries to use IAS while the IAS-I is set, program execution will stop until the interlock is removed.

Tape Control Interlock (TC-I). This interlock is set when any instruction referring to tape is executed (03, 04, 05, 06, 07, 55, 56, 57). The length of time the interlock remains set depends on the operation. Figure 30 shows the various interlock times. While the TC-I is set, the 25 NTS and 54 NEF codes cannot be executed.

FUNCTION	1 Total Cycle Time	2 Program Stopped	3 IAS Interlocked	4 Tape Unit Interlocked	5 Tape Control Interlocked
533 Read	300 ms	61 ms av.			
533 Punch	600 ms	61 ms av.			
537 Punch	387 ms	60 ms av.			
537 Punch	387 ms	90 ms av.			
537 Conditional Read/Punch	387 ms	90 ms av.			
407 Read	400 ms	*			
407 Print	400 ms	*			
407 Conditional Read/Print	400 ms	*			
Tape Read	**		**	**	**
Tape Write	**		**	**	**
Backspace Tape	**			***	**
Read Check Tape	**			**	**
Write Tape-Mark	11 ms			11 ms	11 ms
Rewind Tape	1.2 min			1.2 min	35 ms

*See 650 Programming Bulletin 6, Form 320-7990
**Varies with length of record.
***Varies with length of record plus 51 ms if 57 BST follows a 04 RTN—05 RTA, or 73 ms if the BST follows a 07 WTA, 06 WTN, 56 WTM operation.

EXPLANATION OF HEADINGS
1 The indicated function can only be repeated after the shown interval.
2 No further instruction execution can begin, after the associated function command, for the time shown. Any operations already in progress will complete.
3 No instruction referring to or using IAS can be executed during this time.
4 The specific tape unit cannot be used for other operations during this time.
5 No operations involving other tape units can begin during this time.

Figure 30. 650 Interlock Times

When a 55 RWD command is executed, the TC-I is removed before the operation is finished. This permits tape operations on other units to progress concurrently with a rewind operation.

On any tape command the program is free to continue processing 5-word times after the tape command is found (n + 5 is the optimum factor for the I-address of any tape command). This is approximately one-half millisecond. Figure 30 shows that the TC-I is removed in 35 ms on a 55 RWD command. Therefore, the following program is inefficient because the TC-I prevents the second instruction from being executed for 35 ms.

LOCATION OF INSTRUCTION	INSTRUCTION OP	DATA	INSTRUCTION	OPERATION ABBRV.
0001	55	8010	0006	RWD
0006	04	8011	0011	RTN

About 21 optimized instructions (no referral to tape) can be executed between these two commands with a resultant increase in efficiency.

Placement of the 25 NTS and 54 NEF instructions in the program is important. They should be used to test after every read or write tape command. However, if they immediately follow the read or write tape command, such as in the next program, the result is inefficient because the 25 NTS cannot be executed until the read or write tape operation is completed.

LOCATION OF INSTRUCTION	INSTRUCTION OP	DATA	INSTRUCTION	OPERATION ABBRV.
0001	04	8010	0006	RTN
0006	25	0010	0011	NTS
0011	54	0015	0016	NEF

If we assume that the tape record being read is 400 digits (40 words) long, it takes about 38 ms to read into IAS. Therefore, about 24 optimized instructions (no referral to tape or IAS) can be executed between the 04 RTN and 25 NTS instructions.

Tape Unit Interlock (TU-I). One of these interlocks is available for each tape address (8010-8015). The particular interlock is set when the associated tape unit begins an operation. Its purpose is to prevent the use of a particular tape unit until the current operation on that tape unit is completed. If another instruction referring to the same unit is encountered while the interlock is on, (TC-I OFF), the execution of the instruction will be delayed until TU-I is turned off.

An inspection of the times in Figure 31 shows that the TC-I is turned off before the TU-I only for the rewind and backspace operations. For any other tape operation the TC-I and TU-I are turned off at the same time.

Sequence Charts

The sequence chart (Figure 31) shows the setting and release of IAS-I, TC-I, and TU-I by the associated operation codes.

Operating Pointers

These points should be considered whenever a tape unit is in operation:

1. Do not change the address of a tape unit during the execution of a program which uses other tape units. This applies whether the unit is in ready status or not.
2. Two tape units should never be set to the same address.
3. The door of a tape unit should not be opened unless the tape inside is out of the vacuum columns and the read-write head is raised.
4. In the event of a power failure with tape units in ready status, the tape must be removed from the read-write head and vacuum columns of every unit in a ready status, by an IBM Customer Engineer, before power is restored. Unless removed, extraneous noise may be transmitted to the tapes when power is turned on.
5. Do not turn off DC in the IBM 650 with the tape units in a ready status, as extraneous noise may be transmitted to the tapes when DC is turned on.

Tape Labels

The first record of each reel of tape should be an identification record. This record is used to insure that the proper reel of tape is placed on the correct tape unit in the right sequence for a given job.

The identification record should contain information such as:

1. Serial number
2. Date placed in service
3. Job identification
4. Reel number of job number
5. Tape unit
6. Number of times passed
7. Last date passed
8. Purge date (date when this reel may be used for new data)

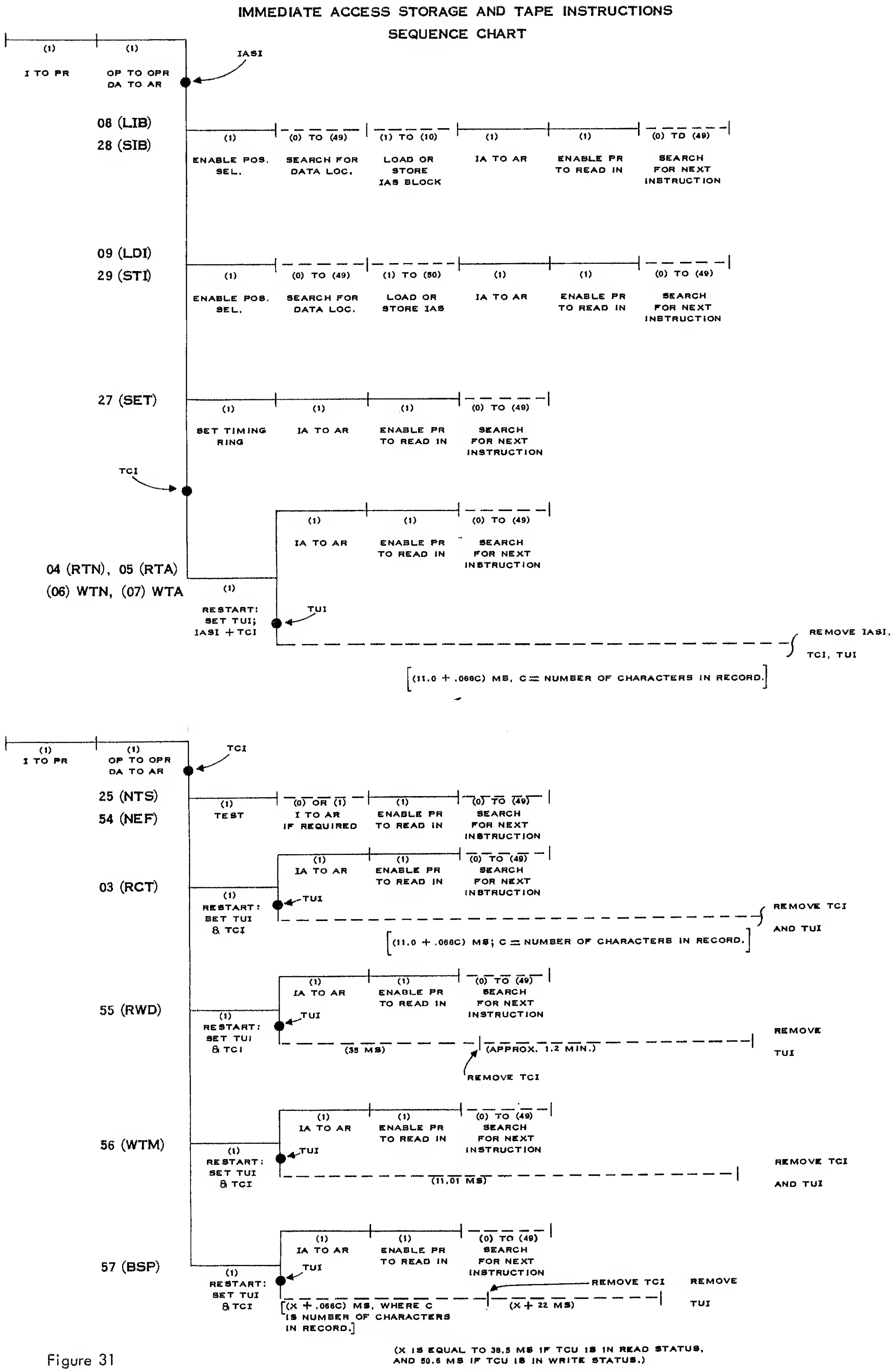


Figure 31. IAS and Tape Instructions Sequence Chart

Tape Load Procedure

When loading a tape reel in the IBM 727 Magnetic Tape Unit, these steps should be carefully followed:

1. Do not remove a tape reel from its plastic container until ready for loading. The plastic container should be closed immediately after removing the tape reel and placed in a location where it is not exposed to dust and dirt.
2. Determine whether the file protection ring is on or off the reel to be loaded, as indicated by the operating instructions to be performed.
3. Open the tape unit door.
4. Unwind about 3½ feet of tape from the reel to be loaded and let it fall free but without touching the floor.
5. Place the reel on the left-drive shaft, with the side on which the file protection ring is located facing toward the tape unit. Push the reel firmly against the stop on the mounting hub so that it is properly seated and aligned. Then tighten the locking hub securely (Figure 32).
6. Pass the tape around the large left-hand roller, across and under the read-write head cover, and then around the large right-hand roller.
7. Place the end of the tape inside the second reel; be sure that it is correctly aligned to prevent damage to the edges of the tape during the first few turns of the reel.

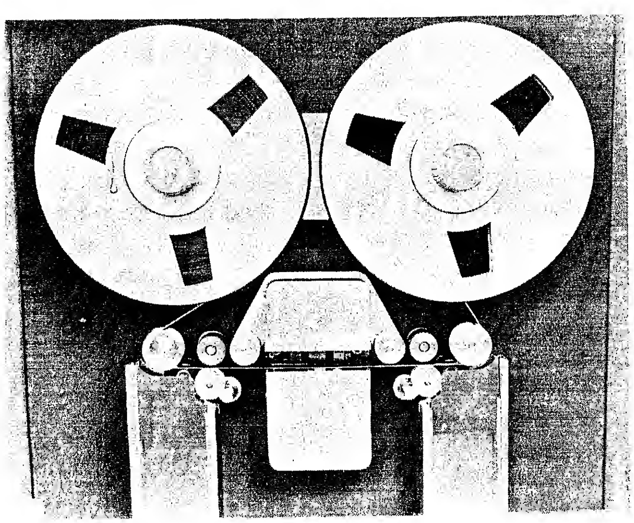


Figure 32. Tape Feeding Mechanism

8. Press both the right and left drive shaft clutch buttons and wind the right-hand reel clockwise (use reel finger hold for winding) until the load point reflective spot passes the read-write head. Continue to wind until the spot is at least two full turns on the right-hand reel. Note: some machines have only one shaft clutch button.
9. Close the tape unit door.
10. Press the LOAD-REWIND key to position the tape loops in the vacuum columns. This causes the tape unit to search for the load point and position the tape for use.
11. At any time, after pressing the LOAD-REWIND key, the start key may be pressed so that the ready light turns on when the load point is found. CAUTION: Pressing the UNLOAD key while tape is being loaded into the vacuum columns can cause the tape unit to blow a fuse.
12. Set the address selection switch to the desired address and make certain that no other tape unit is set to the same address.

Tape Unload Procedure

When unloading a tape unit, these steps should be carefully followed:

1. Press the reset key on the tape unit to return the unit to manual control, and to turn off the ready light.
2. Press the load-rewind key. This causes the tape to rewind. Reels are rewound at an average of 500 inches per second, a full reel is rewound in an estimated 1.2 minutes.
3. Press the unload key. This causes the read-write head to be raised and brings the tape out of the vacuum columns. CAUTION: Do not press the unload key while the tape unit is rewinding. This may cause the tape to break. Also, do not press the load-rewind key while the tape is being unloaded from the vacuum columns. This action may cause the tape unit to blow a fuse.
4. Open the tape unit door.
5. Press both the right and left shaft drive clutch buttons and turn the left-hand reel counterclockwise to wind the remainder of the tape on the reel.
6. Loosen the locking hub and remove the left-hand reel from the tape unit.
7. The sponge rubber grommet should be placed in the reel to prevent the tape from unwinding.
8. Check the removed reel to determine whether it is to be file-protected and whether it has been labeled correctly. Place the reel in the plastic container immediately.

Magnetic Tape Handling

Dust Prevention

Foreign particles on tape can reduce the intensity of reading and recording pulses by increasing the distance between the tape and the read-write head. Be extremely careful to protect magnetic tape from dust and dirt.

Keep the tape in a dust proof container whenever it is not in use on a tape unit. When a reel of tape is removed from a tape unit, immediately place it in a container. Always place sponge rubber grommets or special clips on the reels as they are stored, to prevent the free end from unwinding in the container.

While the tape is on the machine, close the container; put it in some location where it is not exposed to dust and dirt.

Store tapes in a cabinet elevated from the floor and away from sources of paper or card dust. This should minimize the transfer of dust from the outside of the container to the reel during loading or unloading operations.

Never use the top of a tape unit as a working area. Placing materials on top of the units exposes them to heat and dust from the blowers in the unit. It also interferes with the cooling of the tape unit.

Damage Prevention

Information is recorded within .020 inch of the edge of the tape. Proper operation requires that the edge of the tape be free from nicks and kinks.

Handle reels near the hub whenever possible. In picking up reels, grip the reel between the center hole and the outer edge. Gripping the reel so as to compress its outer edges pinches the few turns of the tape near the outer edge of the reel. Personnel handling tape reels inside and outside the machine room should be instructed to avoid pinching the reels or having contact with the exposed edges of the tape.

Dropping a reel of tape can easily damage both the reel and the tape. Never throw or mishandle reels even while they are protected in their containers.

Cleaning Tape and Tape Containers

To clean a tape, gently wipe the tape with a clean, lint-free cloth moistened with tape transport cleaner (customer engineering item). The use of carbon tetrachloride and of Vythene* (cleaning fluid) for cleaning tape should be avoided under all circumstances.

Inspect containers periodically. Remove any accumulation of dust by washing with a regular household detergent.

*Vythene is a registered trademark of Tect, Inc.

Tape Break

If a tape break occurs, divide the reel into two smaller reels. It may be necessary to make a temporary splice in order to recover information; however, splicing is not recommended as a permanent correction procedure. In making a temporary splice, be sure to use the special low cold flowing splicing tape (customer engineering supply item).

Dropped-Tape Inspection

If a reel of tape has been dropped, the reel can be broken or bent (bending is less likely, as a strain sufficient to bend a reel usually breaks it), the edge of the tape may be crimped, and the tape may be soiled. To test for and remedy these defects, proceed as described below:

1. Inspect the tape reel immediately. Breaking or bending of the reel can usually be found by visual inspection. In addition, check the reel for bending by mounting it on the hub of a tape unit. If the reel has been bent or broken, it obviously should not be used again but the tape may be serviceable.
2. Inspect the tape.
   a. If there is no evidence of crimping or other tape damage, and the reel is undamaged, thoroughly clean the tape (exposed or unwound) and reel. The tape is then in good operating condition. If at all possible, test to verify that the tape operates properly before using it on subsequent runs.
   b. If there is no evidence of tape damage, but the reel is damaged, thoroughly clean the tape (exposed or unwound) and rewind it on another reel. If possible, test to verify that the tape operates properly.
   c. If the edge of the tape is crimped, the action to be taken depends on whether or not the tape contains essential information. If the tape does not contain essential information, discard the crimped footage. If the tape contains essential information, thoroughly clean the tape and attempt to reconstruct this information through a tape-to-printer or other machine operation. Should reconstruction fail, the records in question must be rewritten from cards or from another source.

Shipping Tapes

When shipping magnetic tapes, follow these shipping instructions. If a tape arrives from the manufacturer in unusable condition, ship it back to the factory according to these instructions to avoid additional defects before inspection.

1. Place the reel of tape securely in a dust-proof container which supports the tape reel at the center.

2. Hermetically seal the container in a moisture-proof plastic bag.

3. Provide additional support by packing the container in an individual stiff cardboard shipping box.

Storing Tapes

1. Provide adequate protection and mechanical support for the reels of tape by using individual dust-proof containers.

2. Keep the atmosphere of the place of long-term storage of Mylar* tape within the following limits:

Temperature	40-120° F
Relative humidity	0.80%†

3. If the tape is removed from the stated conditions for more than four hours, it must be reconditioned before it is used. Allow it to remain at the given conditions for a length of time equal to the time it was away, up to 24 hours.

†The upper limit is specified to prevent the formation of fungus and mold growth, and can be eliminated by hermetically sealing the tape in a plastic bag.

*Mylar is a registered trademark of Dupont.

Keys and Lights

IBM 727 Magnetic Tape Unit (Figure 33)

Reset Key

Use of this key turns off the ready light if it is on. If the reset key is used during high-speed rewind, the operation stops and then continues as a slow-speed rewind. If the reset key is used during a slow-speed rewind, the rewind stops.

Load Rewind Key

This key is operative only when the reel door is closed and the ready light is off. Use of this key after tape has been properly mounted in the magnetic tape unit lowers tape into the columns, lowers the head assembly, and moves tape in the rewind direction until the load point marker is sensed.

Use of the key with tape loaded and the machine reel containing more than one-half inch of wound tape initiates a high-speed rewind operation. The tape is removed from the columns, the head assembly is raised, and tape is rewound at high speed until about one-half inch of wound tape remains on the machine reel. Tape is then lowered into the columns, the head assembly is lowered, and a low-speed rewind continues until the load point marker is sensed.

Use of the key with tape loaded and the machine reel containing less than one-half inch of wound tape initiates a low-speed rewind until the load point marker is sensed.

CAUTION: Do not open the reel door during rewind or load point searching.

Unload Key

This key is operative only when the ready light is off, tape is in the vacuum columns, and the reel door is closed. Use of this key will remove the tape from the columns and raise the head assembly, regardless of the distribution of tape on the two reels. If the tape is not at load point when the operator wishes to change it, a load-point search should be initiated first by depression of the load-rewind key. Depression of the unload key also turns off the tape-indicate light.

Start Key

This key places the tape unit in ready status and turns on the ready light provided that:

1. The reel door is closed.
2. Tape has been loaded into the columns.
3. The tape unit is not in the process of finding the load point (rewind or load point operation).

Select Light

The select light indicates the last tape unit used in the 650 system.

Ready Light

When this light is on, it indicates that the tape unit is ready for operation. See *Start Key* for method of turning this light ON. The reel door should never be opened when the ready light is ON.

File Protect Light

This light automatically turns on if the unit is loaded with a reel which does not have the file protection

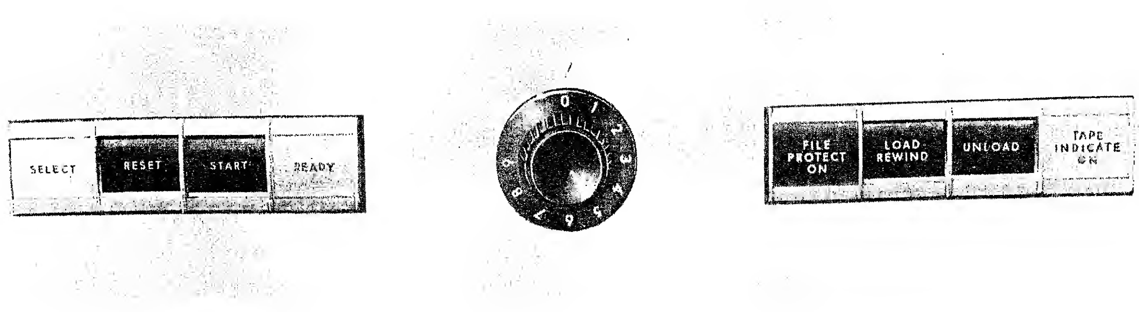


Figure 33. IBM 727 Keys and Lights

ring inserted in the back of the reel. The tape cannot be written as long as the file protection light is ON. If the file protection ring has been removed, and this light fails to go ON, notify the customer engineer immediately. This light is ON during a rewind condition.

Tape Indicate Light

This indicator may be turned on by the following:

1. Sensing the end-of-reel marker while writing on tape.
2. Sensing the tape mark while reading tape.

The indicator may be turned off by depressing the unload key on the tape unit or by selecting the tape unit and branching on an NEF (54).

Select Switch

This switch sets the address of the tape unit to be used. The valid settings are 0-5. No two tape units should ever have the same setting. The machine will stop when a particular tape unit is called for, and the switch has not been dialed to one of the allowed settings (0-5).

IBM 652 Console Display Lights (Magnetic Tape Section, Figure 34)

Selected Lights

The six lights of this section indicate the number of the tape unit in select status.

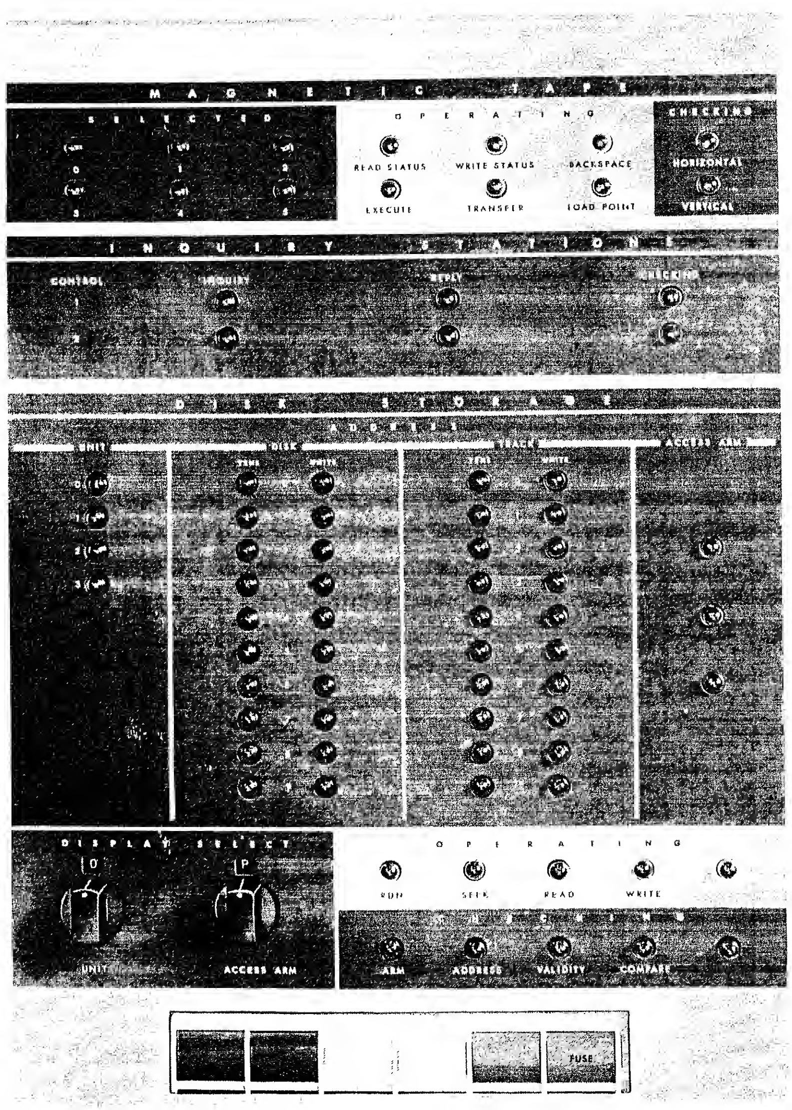


Figure 34. IBM 652 Console Display Lights

Operating Lights

The read status and the write status lights indicate whether the 652 is in read or write status. The read status light is turned on by a read tape (numerical or alphanumerical), a read tape for checking, a rewind, or a backspace instruction. The write status light is turned on when a write tape (numerical or alphanumerical), or a write tape mark instruction is executed by the 650.

The backspace light is on only during the backspacing of a tape. The execute light is turned on during the time the 652 is transferring information from IAS to tape or from tape to IAS. The load point light is on when the tape unit in select status is at load point.

Checking Lights

The appropriate checking light is turned on whenever a horizontal or vertical redundancy error is sensed.

Ready light is on when DC power is on.

Fuse light indicates a blown fuse.

IBM 653 Console Display Lights (Figure 35)

Operating Lights

MAGNETIC TAPE light flashes when any tape unit is being used or referred to.

IAS light flashes when magnetic core storage is being used.

Checking Lights

INDEXING REGISTER light indicates an error in any one of the indexing registers.

IMMEDIATE ACCESS STORAGE light indicates an error in the IAS unit.

Position Lights

These lights are used to indicate the position of the timing ring in the IAS unit.

Each of the three banks of lights (block, word, and digit) serves a separate purpose:

1. BLOCK-6 lights labeled 9000-9010-9020-9030-9040-9050. Indicate the block where the timing ring is located.

2. WORD-10 lights labeled 0-9. These indicate the word within the block where the timing ring is set.
   Block 9020 Word 1 = 9021
   Block 9040 Word 5 = 9045
   Block 9030 Word 9 = 9039

3. DIGIT-10 lights labeled 1-10 used by customer engineers.

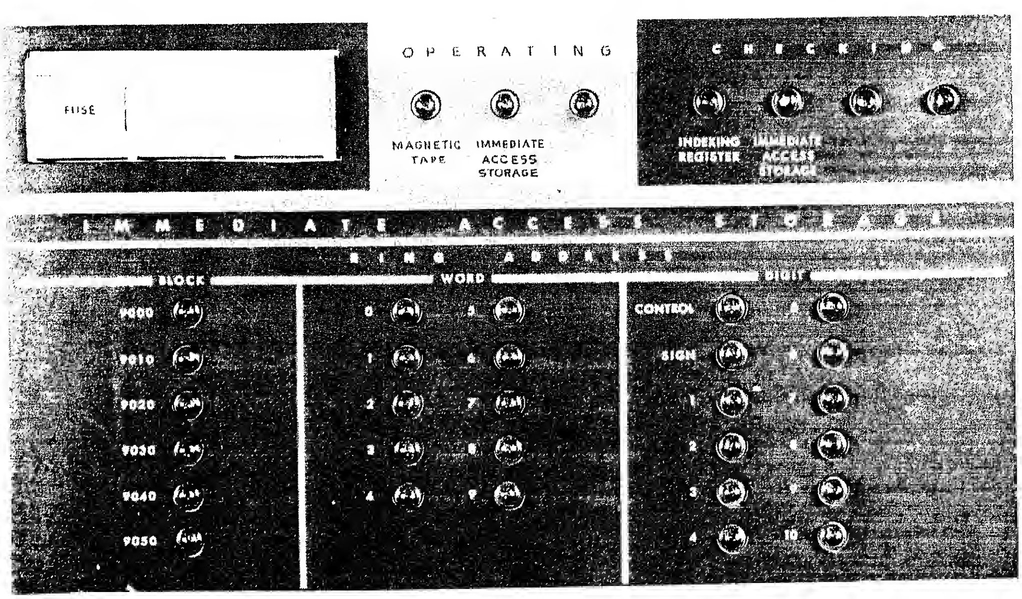


Figure 35. IBM 653 Console Display Lights

IBM 650 Console Display Lights

Control Unit Checking Light (Figure 36)

This light indicates trouble in the magnetic tape operation, and directs the operator's attention to the 652 console.

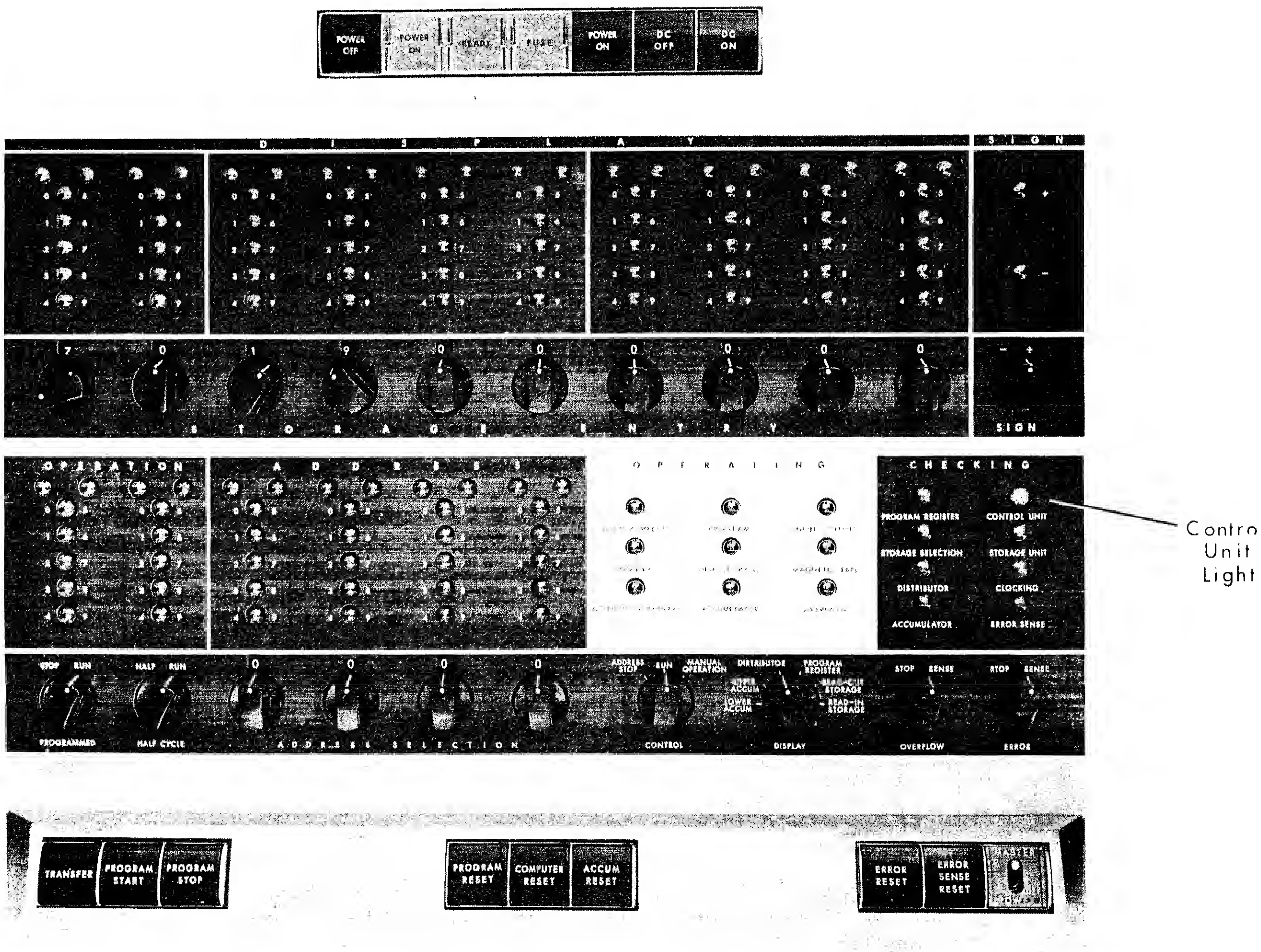


Figure 36. IBM 650 Console Display Lights

112 East Post Road. White Plains. New York
Printed in U.S.A. G24-5003-0 11/59

Index

Address Modification ... 10
Alphabetic Control Word ... 32
Alphabetic or Alphanumerical Mode—Magnetic Tape ... 31
Arithmetic Operations Between Indexing Registers ... 13
Automatic Floating-Decimal Arithmetic ... 20
Automatic Floating-Decimal Arithmetic Device ... 2

Backspace ... 34
Branch No End of File ... 34
Branch No Tape Signal ... 34

Character Coding on Tape ... 28
Cleaning Tape and Tape Containers ... 41
Console Lights—IBM 650 ... 46
Console Lights—IBM 652 Control Unit ... 44
Console Lights—IBM 653 ... 45
Control Unit IBM 652 ... 2

Damage Prevention—Magnetic Tape ... 41
Developed Addresses ... 12
Dropped Tape Inspection ... 41
Dust Prevention—Magnetic Tape ... 41

End-of-File Routine ... 35

File Protection ... 30
Floating Add ... 21
Floating Add Absolute ... 21
Floating Decimal Number in Conjunction with Other 650 Operations ... 25
Floating Decimal Operation Codes ... 20
Floating Divide ... 24
Floating Fixed Decimal Numbers ... 25
Floating Multiply ... 23
Floating Subtract ... 21
Floating Subtract Absolute ... 21

IAI (Immediate Access Impulse) ... 8
IAS with Input/Output Units ... 7
IBM 652 Control Unit ... 4
IBM 653 Storage Unit ... 4
Immediate Access Storage ... 2, 5
Immediate Access Storage Interlock ... 36
Indexing Operation ... 12
Indexing Registers ... 2, 10
Indexing Register Addresses ... 10
Indexing Registers Optimum Timing ... 11
Instruction Indexing ... 11
Interlock Sequence Chart ... 38
IR as Accumulators ... 13
IR Addresses 8005, 6, 7, use of ... 18
IR Arithmetic Operations ... 13
IR Arithmetic Operation Codes ... 15
IR Branching Operations ... 15
IR as Data Address ... 19
IR Operation Codes ... 13, 14, 15

Key and Lights—IBM Magnetic Tape Unit ... 43

LDI (09) Load IAS ... 6
LIB (08) Load IAS Block ... 6

Magnetic Tape ... 27
Magnetic Tape Checking ... 29
Magnetic Tape Checking Impulse (MTC) ... 36
Magnetic Tape System ... 28
Magnetic Tape Unit—IBM 727 ... 2, 27
Mantissa ... 20
Meaningful Address ... 12
Meaningful Data Addresses ... 19
Miscellaneous Tape Operating Codes ... 34
Modified Characteristic ... 20
Multiple-Word Transfer Instructions ... 6

Numerical Mode—Magnetic Tape ... 30

Operating Pointers ... 38
Optimizing Floating-Decimal Operations ... 25
Optimum Programming Using 90xx Addresses ... 8

Photo-Sensing Markers ... 29
Programming for NTS and NEF ... 34

Read Codes 70, 72, 73, 75, 76, 78 ... 7
Read Tape Alphanumerical ... 32
Read Tape Check ... 34
Read Tape Numerical ... 31
Rewind (55 RWD) ... 34

Selection of Tape Read/Write Mode ... 33
SET (27) Set IAS Timing Ring ... 5
Shipping Tapes ... 41
SIB (28) Store IAS Block ... 6
Single-Word Transfers ... 5
STI (29) Store IAS ... 6
Storage Unit—IBM 653 ... 2
Storing Tapes ... 42
Specifications, IBM 727 ... 28
Summary of IR Arithmetic Codes ... 17
Summary of Units in a 650 System ... 2

Table Lookup ... 8
Tagging ... 10
Tape Control Interlock ... 36
Tape Error and End-of-File Conditions ... 34
Tape Error Punch or Print Out ... 35
Tape Error Routine ... 35
Tape Handling ... 41
Tape Labels ... 38
Tape Load Procedure ... 40
Tape Reading Numerical ... 31
Tape Unit Interlock ... 38
Tape Unload Procedure ... 40
Tape Writing Numerical ... 31
Timing Considerations ... 36
Timing Ring ... 5
Transfer of Data to and from IAS ... 5

Unfloating-Floating-Decimal Numbers ... 25
Unnormalized Floating Add ... 21

Validity Check ... 5

Write Codes 71, 74, 77 ... 7
Write Tape Alphanumerical ... 32
Write Tape Numerical ... 31
Write Tape Mark ... 34

International Business Machines Corporation

Data Processing Division

112 East Post Road, White Plains, New York

Printed in U.S.A. G24-5003-0 11/59